AFRL-HE-WP-TR-2004-0169

United States Air Force Research Laboratory

INTEGRATE 2.8: A New Generation Three-Dimensional Visualization, Analysis, and Manipulation Utility

Dennis B. Burnsides

Advanced Information Engineering Services
A General Dynamics Company
5200 Springfield Pike, Suite 200
Dayton OH 45431

June 2004

Interim Report for the Period April 2002 to June 2004

Approved for public release; distribution is unlimited.

Human Effectiveness Directorate
Biosciences & Protection Division
Biomechanics Branch
2800 Q Street, Bldg 824, Rm 206
Wright-Patterson AFB OH 45433-7947

NOTICES

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2004-0169

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE DIRECTOR

// SIGNED //

MARK M. HOFFMAN
Deputy Chief, Biosciences and Protection Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE (DD-MMM-YYYY)
June 2004

2. REPORT TYPE
Interim Report

3. DATES COVERED (From – To)
April 2002 - June 2004

4. TITLE AND SUBTITLE
INTEGRATE 2.8: A New Generation Three-Dimensional Visualization, Analysis, and Manipulation Utility

5a. CONTRACT NUMBER
F33615-02-C-6000

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER
7184

5e. TASK NUMBER
02

5f. WORKUNIT NUMBER
07

6. AUTHOR(S)
Dennis B. Burnsides

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Advanced Information Engineering Services, A General Dynamics Co.
5200 Springfield Pike, Suite 200
Dayton OH 45431

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Human Effectiveness Directorate
Biosciences & Protection Division
Biomechanics Branch
Air Force Materiel Command
Wright-Patterson AFB OH 45433-7947

10. SPONSOR / MONITOR'S ACRONYM
AFRL/HEPA

11. SPONSOR/MONITOR'S REPORT NUM BER(S)
AFRL-HE-WP-TR-2004-0169

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report documents the functionality available using INTEGRATE 2.8, a Linux or Windows-based software package, to visualize, analyze, and manipulate three-dimensional topographic data. The analysis capability represented by this software is robust, flexible, and instrumental in applying 3-D anthropometry toward the improved fit of protective equipment, clothing, commercial head gear, and medical devices. Tutorials are available to guide the user through representative applications.

15. SUBJECT TERMS
Surface Scanning, WB4, Visualization, Manipulation, Anthropometry, Human Body Topography, Image Analysis

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON:
a. REPORT	b. ABSTRACT	c. THIS PAGE			Kathleen M. Robinette
U	U	U	SAR	184	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-8810

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39-18

This Page is Intentionally Left Blank

PREFACE

This research was conducted by the Computerized Anthropometric Research and Design (CARD) Laboratory of the Biosciences and Protection Division, Human Effectiveness Directorate, Air Force Research Laboratory, Wright-Patterson Air Force Base, Ohio. The work was performed under the Advanced Anthropometric Accommodation (AAA) Contract Number F33615-02-C-6000. The author is grateful for the support of all personnel in the CARD Lab, especially its director Kathleen Robinette, and is indebted to the authors of reports describing earlier stages of this work.

This Page Is Intentionally Left Blank

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

This Page Is Intentionally Left Blank

INTEGRATE 2.8: A New Generation Three-Dimensional Visualization, Analysis, and Manipulation Utility

1.0: INTRODUCTION

1.1: Background

The Computerized Anthropometric Research and Design (CARD) Laboratory, at Wright-Patterson Air Force Base, Ohio has been using surface scanning technology to improve equipment design applications since 1987. CARD Lab researchers continue to evaluate many commercial software packages, such as Computer-Aided Design (CAD), to determine their utility for manipulating surface data for USAF equipment designs. These software programs, however, proved incapable of providing the unique functions required to analyze topographic data on people and their equipment. For this reason, the CARD Lab developed INTEGRATE as a prototype system to test the functionality required to visualize, analyze, and manipulate surface data. The current version of INTEGRATE, version 2.8, offers functionality to meet the needs of current USAF engineering anthropometry challenges. With Version 2.8, a number of commands have been added and versions are available for IRIX, Linux, and WIN32. In addition, syntax has been added to enable limited object manipulation based on landmark positions. Researchers from businesses and universities worldwide use earlier versions of INTEGRATE to test and evaluate new helmet systems, develop augmentative files such as landmark data or contour information for surface scan databases, record human-equipment interface geometries, extract measurements equivalent to traditional anthropometry for whole body image data, and prepare surface data for rapid prototyping systems (e.g. Perkins and Blackwell, 1998; Daanen et al., 1996; Brunsman et al., 1996; Brunsman et al., 1996; Robinette et al., 1994; Robinette et al., 1992; Whitestone et al., 1995; Whitestone et al., 1993; Whitestone et al., 1992).

Because INTEGRATE was designed as a prototype, user friendliness was not a high priority for the developers. However, a few hours experience with the program and this document should familiarize the user with commonly used commands and the general architecture of the software. This document contains six sections:

1. Introduction,
2. General Operating Instructions,

3. Tutorials,
4. INTEGRATE Command Reference,
5. INTEGRATE's Audit Trail Function, and
6. Appendices.

The user should read the General Operating Instructions before beginning the Tutorials. It is highly recommended that the user "walk through" at least the first two tutorials before beginning a new session. There are tutorials that are targeted for users of both head scans and whole body surface data. The INTEGRATE commands are organized in alphabetical order with examples of the use of each command. Finally, the Appendices are included to provide the user with additional information such as anatomical landmark definitions and illustrations, listings of files needed for the tutorials, and script files for routine sessions.

1.2: Functionality

The goal in the design of INTEGRATE is to provide for future functions so that no changes in the basic program, functions, and data structures will be needed to add any new function. New functions can and will be added quickly when the need arises.

1.3: Object Pool

The Object Pool keeps track of all the information for each object. INTEGRATE can work with an arbitrary number of objects at one time (the present limit is 100). A rule of thumb is that approximately 13 million points (100 head scans) can be in the Object Pool at one time. These objects can be displayed or hidden by the user. The amount of memory in use is displayed as a percentage (13 million points = 100%) in the **Global Status Window**. Note that, for historic reasons, only grid objects are used in computing memory used.

2.0: GENERAL OPERATING INSTRUCTIONS

2.1: Starting INTEGRATE

INTEGRATE was originally developed on the Silicon Graphics 4D models. While the CARD Laboratory at the time of this publication is using IRIX 6.5, Linux kernel 2.4, and Windows 2000 Professional, INTEGRATE may run on other versions of all of these operating systems. INTEGRATE Version 2.8 is now available for IRIX, Linux, and WIN32 operating systems. To start the INTEGRATE program, login to the computer system, then type **integrate** or click on the IntegrateWin icon. The screen will show the X and Y axes, and will indicate **No Active Object** in the Active Object Status Window in the lower left corner of the screen.

2.2: INTEGRATE Status Windows

INTEGRATE has 3 status windows across the bottom of the screen:

the **Active Object Status Window**,

the **Object Summary Window**, and

the **Global Status Window**.

<u>2.2.1: The Active Object Status Window</u>, located in the lower left corner of the screen, contains information about the current Active Object. Many INTEGRATE commands operate on the Active Object, so it is important to view the current status of an object before modifying it. Figure 1 provides an example of the Active Object Status Window.

```
SubjFile: 101_53p
LandFile: 101_53p.lnd
Active:5   Lon Thin:1   Lat Thin:1
Left:0  Right:512  Lower:0  Upper:256
Angles: X:   13.4  Y:   72.9  Z: 357.3
Center: X:    0.0  Y: 200.1  Z:    0.0
Offset: X:   -4.2  Y: -61.9  Z:   47.3
```

Figure 1: Active Object Status Window

The following information appears in the **Active Object Status Window**:

- SubjFile: the name of the file containing the original data points (or the name of the geometry file last written from this object).
- LandFile: the name of the file containing the landmark points (or the name of the landmark file last written from this object).
- Active: the number of the Active Object (this object).
- Thin Factors: the number of longitudes and latitudes INTEGRATE skips when displaying the object (grid objects only).
- Corners: the Left and Right longitudes and the Lower and Upper latitudes of the subsection of the active object (grid objects only).
- Angles: the X, Y, and Z rotation angles from the original object position to the displayed object position.
- Center: the X, Y, and Z offsets to center the object in the axis system.
- Offset: the X, Y, and Z offsets to move the object from its original (centered) position to its displayed position.

2.2.2: The Object Summary Window, located in the lower right corner of the screen, lists every object currently loaded into the Object Pool, its file name, and its display status. This window is color-coded to help determine which image in the display area is associated with which object. Figure 2 provides an example of the Object Summary Window.

1	53psize5	+W1		7	102_53p	WS1
2	100_53ph	S1		8	104_53ph	W1
3	100_53p	W1		9	104_53p	
4	101_53ph			10	105_53ph	W
5	101_53p	* S		11	105_53p	W1
6	102_53ph	W1				

Figure 2: Object Summary Window

Each object's summary appears in this order:

(object number) (subject file name) (display status)

The object number is the number to use to select that object for use in a command. The subject file name helps determine which object is to be selected, and the display status indicates the status of an object.

Table 1 below defines the symbols used in the display status line:

Table 1: Display status line definitions.

Symbol	Meaning
*	This is the active object.
+	This object is on the screen now, not hidden.
W	Wireframe display is on for this object.
P	Point display is on for this object.
S	Surface display is on for this object.
T	The surface display for this object is semi-transparent.
l	Landmark display is on for this object.
c	Contour and circumference display is on for this object.

2.2.3: The Global Status Window is located between the active object status window and the object summary window. The global status window contains information about the INTEGRATE environment, such as eye position, pick mode, and clipping wall locations. Figure 3 provides an example of the global status window:

```
Walls:   100:1400:1300
Eye: X:0 Y:0 Z:700 Dist:700
Data Path: new_tut/
Memory Use: 11% (1408K)
PICK OFF
```

Figure 3: Global status window

The global status window contains the following information:

- Walls: Clipping Wall positions and the distance between them.
- Store: If Store is visible on line 1, measurement storage is enabled.
- RGB: If R is displayed on line 1, RGB (full color) mode is enabled.
- Gouraud: If G is displayed on line 1, GOURAUD shading is enabled in RGB mode.
- Orthogonal View: If O is displayed on line 1, orthogonal view is enabled.
- Eye: Eye position and distance with respect to the center of the coordinate system.
- Data Path: The prefix INTEGRATE adds to a load command file name to locate the file.
- Memory Use: A rough estimate of the percentage of the available memory being used to store grid object information (100 head scans=100%).
- Pick Mode: Pick mode is either OFF or the active pick mode is displayed
- Pick Prompt: In some pick modes, the pick prompt indicates the next desired point in the pick sequence
- Intrplnd: If * is displayed next to pickmode, intrplnd is enabled.

2.3: Operating Features

INTEGRATE has a number of operating features that help the user manipulate displayed data. These include the **echo buffer**, use of the **up and down arrow keys, point picking, the function key menu, script files**, and **toggle options**.

<u>2.3.1: The Echo Buffer</u> is a section in the lower left corner of the screen which displays the commands as they are typed. The area immediately above the echo buffer displays the status of operations in

progress, reports operator errors, and displays command usage information for complex commands. The echo buffer also supports:

- the Home key (go to start of command),
- the End key (go to end of command),
- the Delete key (delete char at cursor),
- the Backspace (<-) key (delete char left of cursor),
- the Insert key (insert a blank at the cursor), and
- the left and right arrow keys (move cursor without changing text).

2.3.2: The up/down arrow keys recall the previous command in the command history list to the echo buffer. This feature is circular; when the oldest available command is displayed, the **up arrow** cycles to a blank line, then repeats starting with the newest command. The **down arrow** key recalls the next command in the command history list to the echo buffer.

2.3.3: Point Picking consists of 3 steps:

1) enabling point picking (PICK ON),
2) selecting a pick mode (PICKMODE), and
3) picking points with the mouse by placing the cursor and clicking the left mouse button.

When pick mode is active, when the left mouse button is pressed the points near the cursor will be picked and processed according to the pick mode. In some pick modes, the center mouse button clears/resets the processing for that mode. For example, in Pick Mode CON3P, if the center mouse button is pressed after the second point is picked, the Pick Mode will be reset to restart CON3P picking, with the next point being used as point 1. The right mouse key brings up a "popup" menu which can be used in place of the keyboard for many of the INTEGRATE commands.

The available function key commands are listed across the top of the screen. These keys are user-configurable through a file called fkey.tbl, which resides in the INTEGRATE directory or the user's current working directory. Either clicking on the command or pressing the function key combination (F=no modifier keys, S=shift key, C=Ctrl key, and A=Alt key) activates the command. There are 2 types of function key commands: complete commands which are immediately executed, and partial commands, which are placed into the echo buffer where the user can add parameters to the command before pressing the Enter key to execute the command.

The INTEGRATE directory is accessed through an environment variable called INTEGRATE. An example command creating this environment variable is: "setenv INTEGRATE /home/code/INTEGRATE". This command can be placed in the user's .login or .cshrc file so that it will be activated when the user logs in. If the INTEGRATE environment variable is not set, the MAN command will not work, since it uses the INTEGRATE variable to find the users_guide.txt file, which contains a text version of this document as well as descriptions of any added functionality since the last time the manual was published. A second environment variable, INTEGRATE_FILES, can be used to access a common directory of INTEGRATE scripts.

2.3.4: The Script Execution Capability gives INTEGRATE much of its power. Because INTEGRATE is command-line-oriented, it can require a lot of repetitive typing to perform even simple tasks. Because of this limitation, and because the CARD Lab needs the capability to perform repetitive operations many thousands of times in some cases, INTEGRATE has been provided with a script execution capability that can reduce the amount of required typing and can completely automate complex tasks so that interaction with a human operator is completely eliminated. With this capability, INTEGRATE is able to completely process large amounts of data without devoting expensive manhours to each dataset.

The INTEGRATE scripting capability uses exactly the same commands as the interactive mode, with some additions that extend the power to use scripts. Any command that can be typed in the echo buffer can also appear in a script. In addition, scripts support a few additional commands, a parameterization ability to generalize scripts, and a syntax for doing limited math on landmark coordinates, as described below.

INTEGRATE scripts are activated either from the echo buffer or when INTEGRATE is started from a shell window. To activate a script from the echo buffer, type "@<script file name> [parameters]". To activate a script from the shell window, type "INTEGRATE <script file name> [script file name] ..." Note that script files specified from the shell window cannot have parameters, while scripts executed from the echo buffer (or other scripts) can. There is no known limit to the ability to "nest" command file activations within other command files, but it is undoubtedly possible to exceed the resources available on a particular system with a badly-designed script file structure (especially a case where script file 1 activates script file 2, which in turn activates script file 1).

2.3.4.1: Four commands are available that are only used in scripts: **pause, jump, refresh, and wait**. **Pause** halts the script until the operator restarts it with a key press or mouse click. **Jump** provides a limited looping capability. **Refresh** forces a screen update, usually appearing just before a **pause** command. The **wait** command enables or disables pauses after the display of error or status messages. When **wait** is off, scripts do not pause after displaying messages.

2.3.4.2: Scripts may be parameterized to make them more universally applicable to data processing needs. For instance, **"pload %1.%2.ply"** is much more useful in a script than **"pload csr0060a.abdomen.ply"**. The first script line could be used with a wide variety of subjects and body segments merely by specifying the desired subject and body segment desired, while the second line could only be used with the abdomen segment of CAESAR subject 60. If the script file is named "loadit", typing **"@loadit csr0060a abdomen"** in the echo buffer or in another script file reproduces the second command. When the script is much more complex than a single line, the value of a parameterized script is even higher.

2.3.4.3: One requirement that keeps recurring is the need to perform some operation on a dataset based on one or more landmark coordinates. Many landmark manipulation commands allow some processing based on landmark positions, but there was a need for a more general capability. This capability has been added to the script processing functions. An example of the need for landmark-based manipulation is segmentation based on landmarks. The **movie_seg** command can be used with landmark coordinates to divide the body into segments. An example command might be: **movie_seg lx(&1Z29X-10) ux(&1Z41X+10) uy(&1Z24Y+10) ly(&1Z25Y-10)**. Assuming a scan in standard position (+X left, +Y up, and +Z front) and the CAESAR landmark set, this command would remove all points more than 10 mm outside the acromions (shoulders) (z29 and z41), above the cervicale (base of neck) (z24), and below the 10^{th} rib midspine (bottom of rib cage) (z25), separating the chest area (and probably some portions of the arms) into an independent object.

The syntax of the script mathematics capability is in Appendix I.

2.3.5: The INTEGRATE Toggle Options enable or disable screen features and operating modes. A toggle option may be set by typing **<option command> on** , cleared by **typing <option command> off**, or toggled by typing only the **<option command>**. Most toggle options are tied to function keys which is where the toggle feature is most valuable.

3.0: TUTORIALS

INTEGRATE supports a great deal of functionality, but it is only with experience that the user will be able to take full advantage of the tools available. The following tutorials lead the user through some common INTEGRATE activities to demonstrate the process used to generate the end result.

The image data and script files needed for these tutorials should be available with INTEGRATE version 2.8. A listing of the necessary files for each tutorial is found in Appendix A.

To see the end result of each tutorial, run the tutorial script files. To run the script file for the first tutorial, Basic Moves, type this command in INTEGRATE:

@tutorial_1.txt

To run the script file for the second tutorial, Registration Techniques, type:

@tutorial_2.txt

Each tutorial has a corresponding script file. To gain experience with the INTEGRATE commands, however, new users should execute each tutorial step by step, without using the script files.
The tutorials are as follows:

Tutorial_1	**Basic Moves**
Tutorial_2	**Registration Techniques**
Tutorial_3	**Point Picking**
Tutorial_4	**Feature Envelopes**
Tutorial_5	**Radial Difference Maps**
Tutorial_6	**Helmet Clearance**
Tutorial_7	**Calipers/Tape Measure**
Tutorial_8	**Establishing Joint Centers on Whole Body Data**
Tuturial_9	**Segmenting Whole Body Data**
Tutorial_10	**Articulating/Re-Posing Segmented Whole Body Data**

The tutorials are presented in a table format. Tutorial steps appear in the left column, and the commands used to carry out each step appear in the right column. Refer to section 4.0 Commands for additional information on how the commands work.

3.1: Tutorial One: Basic Moves

This tutorial introduces the user to the basic commands needed to manipulate the object on the screen. The user will learn to initially position the object, move the eyepoint, turn off and on landmarks, change the representation of the object from wireframe to surface, and other essential functions for visualizing the image. Shown in Figure 4 is the image "010_53p", a scan of an unencumbered subject with blue dots indicating the location of anatomical landmarks (shown here in monochrome).

Figure 4: Scan of an unencumbered subject (shown in monochrome for this publication)

The files needed for this tutorial are:

010_53p
010_53p.rgb
010_53p.lnd

Steps	Commands
1. Enter INTEGRATE and begin session.	integrate (or click on icon)
2. Load the unhelmeted scan file of the subject.	cload 010_53p
3. Load the anatomical landmark file associated with this image and update screen image.	lload 010_53p.lnd
4. Rotate the subject around the Y axis so he is directly facing you. Notice that the positive Y axis value causes counterclockwise rotation. This rotation initially places the object so that the "front" of his face is facing you. This coincides with the default eyepoint of INTEGRATE which is "front". All subsequent eyepoint commands (e.g. "back") will correspond with the object (e.g. back of the head). The axis and amount of rotation will depend on the orientation of the object when scanned.	rotate 0 75
5. Change the eyepoint to view the image from the right. Note that this does not MOVE the object, it only changes your viewpoint.	right (or Shift-F6 or click on "right" in func key menu)
6. Change the eyepoint to view the image from the back.	back (or Shift-F8 or click)
7. Change the eyepoint to view the image from the left.	left (or Shift-F9 or click)
8. Change the eyepoint to view the image from the top.	top (or Shift-F5 or click)
9. Change the eyepoint to view the image from the bottom.	bottom (or Shift-F10/click)
10. Change the eyepoint to view the image from the front.	front (or Shift-F7/click)
11. Move the object 50 mm along the x axis. Notice that the object is moving relative to the screen.	move 50
12. Move the object 50 mm along the y axis. Notice that the object is moving relative to the screen.	move 0 50
13. Change the eyepoint to view the image from the right.	right
14. Move the object 50 mm along the z axis. Notice that the object is moving relative to the screen.	move 0 0 50
15. Move the object back to the original origin.	move -50 -50 -50
16. Change the eyepoint back to the front.	front
17. Trim away extraneous image noise at the top and bottom of the subject.	trim 0 0 0 -55 trim 0 0 55

Steps	**Commands**
18. Make a copy of the object and save it as object #2.	copy
19. Create voids in the image data.	ruin 1 2
20. Hide the second object.	hide 2
Fill in the voids on the image and the gap at the top of the subject's head. Notice that "1" was first selected to perform the operations on the first object.	1 do fill toupee 200 205
22. Turn off the axes and the status windows.	axes (or Shift-F3/click) boxes (or Ctrl-F7/click)
23. Turn the status windows back on and toggle the representation of the landmark locations to crosshairs from "L#" and back (# denotes the landmark number found in the landmark file). See Appendix D.	boxes alt_land (or Ctrl-F5/click) alt_land
24. Turn on the landmark list to view the active landmarks and their coordinates in the object coordinate system.	landlist (or Shift-F11/click)
25. Turn off the landmark list and turn on the help list.	landlist help (or F11/click)
26. Turn off the help list and turn on the function keys listing.	help fkeys (or F7/click)
27. Turn the landmarks off, then back on.	land (or F5/click) land
28. Turn off the points mode and display the surface of the object to show texture and color. Then turn off acquired color to see a shaded representation of the surface.	points (or F3/click) surface (or F2/click) fullcolor 010_53p rgb
29. Calculate the volume and surface area of the object.	volume surface_area

3.2: Tutorial Two: Registration Techniques

This tutorial demonstrates registration techniques used to visualize a subject within a helmet system, as shown in Figure 5. This registration technique can be used for examination of a subject within any protective equipment item. As shown in Figure 5, an "x-ray" view is provided, allowing the designer to look inside the human-equipment interface.

Figure 5: Registration of subject with helmet scan for visualizing subject/equipment interface

Three image files are needed for the registration procedure:

1) one scan file of the subject with at least three visible anatomical landmarks,
2) one scan file of the same subject expertly fitted with a helmet system and showing at least the same three anatomical landmarks plus three reference landmarks on the helmet system, and
3) one scan file of just the helmet system with the same three helmet reference landmarks.

In this example, the landmarks have been identified and saved to a landmark (*.lnd) file for each image file. The *.rgb files are color files associated with each scan. The files used in this tutorial are:

010_53p, 010_53p.rgb, 010_53p.lnd

010_53ph, 010_53ph.rgb, 010_53ph.lnd

53psize5, 53psize5.rgb, 53psize5.lnd

Steps	**Commands**
1. Enter INTEGRATE and begin session.	integrate
2. Load the unhelmeted scan file of the subject.	cload 010_53p
3. Rotate the subject around the Y axis so he is directly facing you. Notice that a positive Y axis value causes counterclockwise rotation.	rotate 0 75
4. Trim away extraneous image noise at the top and bottom of the subject.	trim 0 0 0 -55 trim 0 0 55
5. Fill in voids on the subject and the gap at the top of the subject's head.	do fill toupee 200 205
6. Load the anatomical landmark file associated with this image.	lload 010_53p.lnd
7. Load the helmeted scan file of the same subject with his helmet donned.	cload 010_53ph
8. Trim the noise from this image.	trim 0 0 0 -50 trim 0 0 45 0
9. Load the landmark file associated with this image.	lload 010_53ph.lnd

Steps	Commands
10. Register the helmeted image with the unhelmeted image. Notice that the second image is rotated and translated into the coordinate system of the first image, and that lregister is used to align the scans as the common landmarks are anatomical landmarks.	lregister 2 1
11. Change the viewpoint to view the images from the right.	right
12. Use the walls command to "slice" through the data to examine the alignment of profiles.	walls 695 699 walls +5 walls +5 walls full
13. Change the eyepoint back to the front.	front
14. Load the helmet scan (and landmark file) for registration with the helmeted image file.	cload 53psize5 lload 53psize5.lnd
15. Register the helmet scan with the helmeted image file and view the alignment. Notice that zregister is used to align the scans as the common landmarks are auxiliary landmarks.	zregister 3 2 right walls 698 699 walls +5 walls +5
16. Hide the helmeted scan and show only the subject and the scan of the helmet alone. This final configuration illustrates the position of the subject within the helmet.	hide 2 walls full
17. Change the subject file to a surface and the helmet scan to a wireframe of lower resolution.	1 points surface 3 points wireframe thin 2 2
18. Change the helmet scan to a transparent surface.	wireframe surface transparent

Steps	Commands
19. Change the subject file to represent the color information.	1 fullcolor 010_53p
20. View this configuration from different viewpoints.	front left back front

3.3: Tutorial Three: Point Picking

This tutorial demonstrates how to access and implement the point picking capability to generate a landmark file for the scan data. In this case, a head scan is loaded into INTEGRATE and the anatomical landmarking sequence initiated. This is a canned landmarking sequence that includes 42 head and face anatomical landmarks. The landmark selection order, shown in the global status window, has been established to allow the user to begin landmark selection on the right side of the head and progress around the head, working from top to bottom. The actual landmark file, however, lists the landmarks not in the order of selection, but in the order found in Appendix D. If a new landmark *picking order* is required, the command **new_order** can be used to establish a order for picking the points. Figure 6 shows a head scan file with anatomical landmarks.

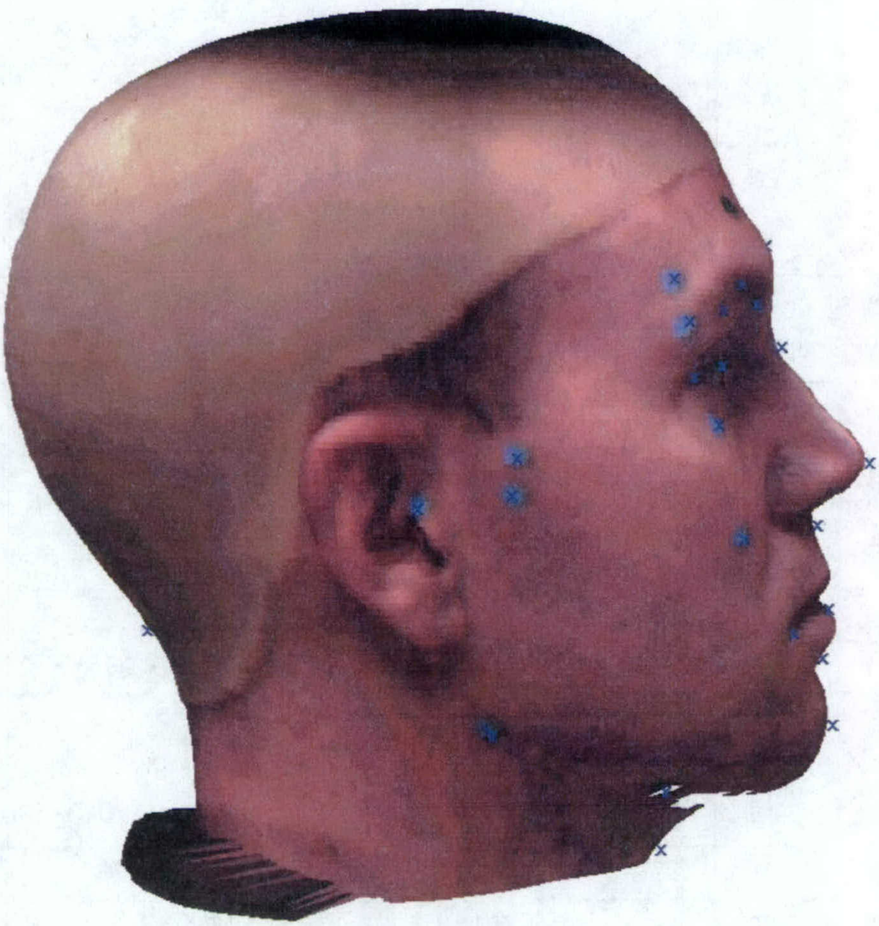


Figure 6: Subject scan with color file (monochrome for this publication) and marked landmark locations

User-defined or arbitrary reference landmarks can be selected and stored in the landmark file using the command **pickmode aux_land.** A landmark file format is found in Appendix D.

Appendix B provides an illustration of the head and face anatomical landmarks. Refer to this figure during the landmarking process. For further clarification, definitions of the landmarks are also included in Appendix B. The landmark to be picked appears in the Global Status Window.

The files needed for this tutorial are:

010_53p
010_53p.rgb
010_53p.lnd

Steps	**Commands**
1. Enter INTEGRATE and begin session.	integrate
2. Load and position the head scan file.	cload 010_53p rotate 0 75
3. Trim away extraneous image noise at the top and bottom of the subject.	trim 0 0 0 -50 trim 0 0 58
4. Fill in voids on the subject and the gap at the top of the subject's head.	do fill toupee 205 207
5. Change the surface from wireframe to color representation. Color mode clearly displays the color landmark dots.	points surface fullcolor 010_53p
6. Move viewpoint to right side to pick right side landmarks.	right
7. Begin landmarking session.	pick on pickmode land

Steps	Commands
8. Use the mouse to put the cursor on the first landmark to be picked, the right tragion (near the inner ear; refer to the diagram in Appendix B). Click the left mouse button to select the landmark. Look in the global status window for the prompt that lists the next landmark to be picked. Pick several more landmarks.	
9. Skip backward in the landmark list to re-pick the last landmark. When the prompt in the global status window changes to the previous landmark name, re-pick that landmark.	skip -1
10. Save the landmark locations to a landmark file.	lwrite land_010.lnd
11. Display the landmark list to view the landmark coordinates.	landlist
12. Turn off the landmark list and turn the function key display back on.	landlist fkeys
13. Compare the selected landmarks with the standard landmark file for the subject. Load the subject file with its landmark file and rotate it into the same orientation as the original subject.	cload 010_53p 010_53p.lnd rotate 0 75
14. Change the original subject to wireframe to compare the landmarks.	1 surface wireframe
15. Change the viewpoint to examine the scans from different views.	front left back front

3.4: Tutorial Four: Feature Envelopes

This tutorial demonstrates how INTEGRATE can be used to generate feature envelopes for equipment items such as a helmet system. Feature envelopes describe the spatial location and orientation of areas of interest (i.e., features) with respect to a well defined, easily duplicated coordinate system. For a given helmet system, this definition could include the range of pupil location along all three coordinate axes or the volume which contains the aggregate of all ears for a given population.

These anthropometric design envelopes defined for an existing helmet are based on one critical factor: the relationship of the head to the helmet. Helmet systems do not fit the human head in exactly the same way across a sample of people. Figure 7 illustrates two subjects wearing the same helmet.

Figure 7: Two subjects wearing the same size and model helmet

The orientation of the head with respect to the helmet system is entirely dependent on the shape of the helmet, the liner system, and the added peripherals, such as optics or earcups. All of these components must be fit optimally to the individual and, as a result, the helmet system "sits" on the head in a slightly different manner for everyone. In order to study these anthropometric design issues, researchers need surface scanning combined with the tools available in INTEGRATE. An example of the pupil envelopes of five subjects for a USAF helmet system is shown in Figure 8.

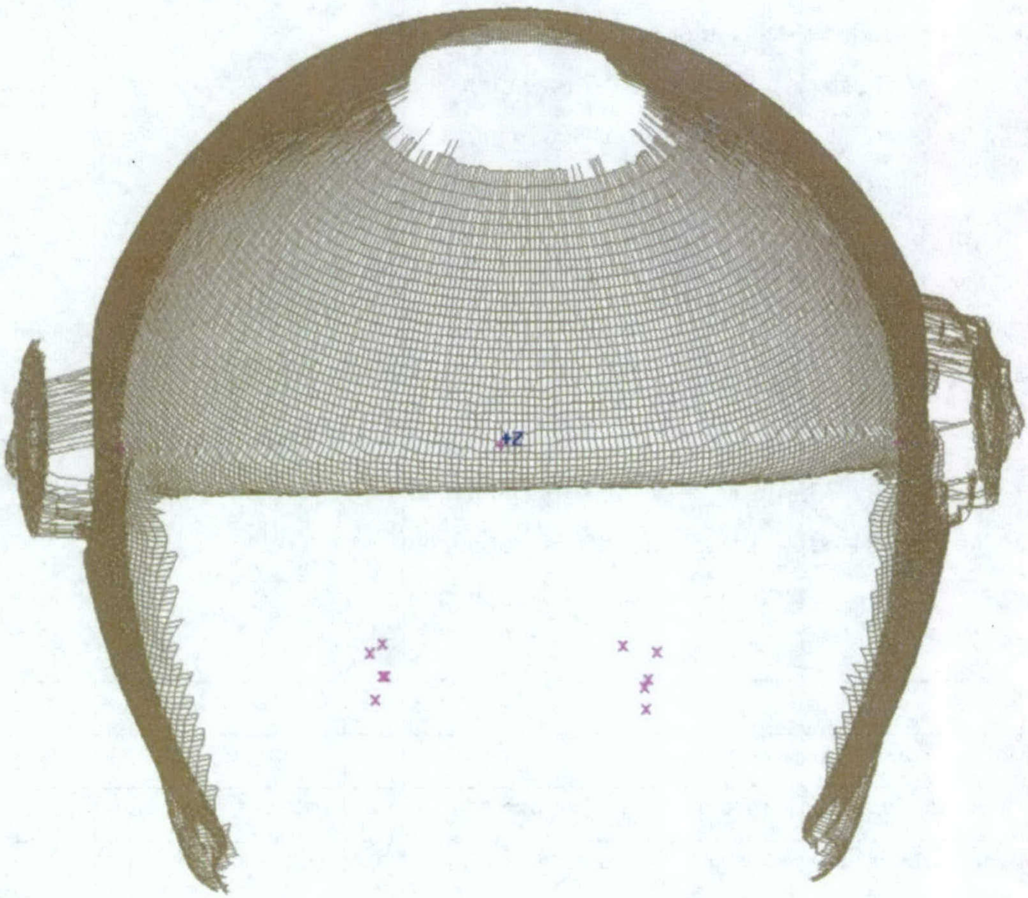


Figure 8: Pupil envelopes for five subjects in the same model and size helmet

As in Tutorial 1, this tutorial consists of aligning an encumbered (helmeted) scan with that of a scan of the helmet alone and aligning the unencumbered (bare head) scan with that of the helmeted scan. This is performed by registration of the helmet landmarks found on the helmet scan with common landmarks found

found on the encumbered scan and registration of anatomical landmarks. The location of the subject can then be viewed with respect to the helmet coordinate system. Specifically, the locations of the pupils for each subject can be determined with respect to the helmet system. This is performed, in this tutorial, for a total of five subjects.

In this example, the landmarks have been identified and saved to a landmark (*.lnd) file for each image file. The *.rgb files are color files associated with each scan.

The following files are needed for this tutorial:

53psize5, 53psize5.rgb, 53psize5.lnd
100_53p, 100_53p.rgb, 100_53p.lnd
101_53p, 101_53p.rgb, 101_53p.lnd
102_53p, 102_53p.rgb, 102_53p.lnd
104_53p, 104_53p.rgb, 104_53p.lnd
105_53p, 105_53p.rgb, 105_53p.lnd
100_53ph, 100_53ph.rgb, 100_53ph.lnd
101_53ph, 101_53ph.rgb, 101_53ph.lnd
102_53ph, 102_53ph.rgb, 102_53ph.lnd
104_53ph, 104_53ph.rgb, 104_53ph.lnd
105_53ph, 105_53ph.rgb, 105_53ph.lnd

Steps	**Commands**
1. Enter INTEGRATE and begin session.	integrate
2. Load the helmet scan with the helmet reference landmarks.	cload 53psize5 53psize5.lnd
3. Rotate the helmet into a helmet-based coordinate system. This coordinate system is based on easily located, symmetric, consistent reference marks on the helmet.	align xz z2 z5 z3 z3
4. Load the first subject's encumbered (helmeted) scan with landmark files containing both helmet reference landmarks and anatomical landmarks.	cload 100_53ph 100_53ph.lnd
5. Register (align) this scan with the helmet scan using the common helmet landmarks.	zregister 2 1

Steps	**Commands**
6. Load the first subject's unencumbered (bare head) scan with anatomical landmarks.	cload 100_53p 100_53p.lnd
7. Register (align) this scan with the encumbered scan. In effect, this aligns the subject with the helmet scan.	lregister 3 2
8. Copy the pupil landmark locations to the helmet scan. This associates the pupil locations with respect to the global helmet system. Then hide the subject scans.	1 copyland l1 3 l34 copyland l2 3 l38 hide 2 hide 3
9. Load the second subject's encumbered (helmeted) scan with landmark files containing both helmet reference landmarks and anatomical landmarks.	cload 101_53ph 101_53ph.lnd
10. Register (align) this scan with the helmet scan using the common helmet landmarks.	zregister 4 1
11. Load the second subject's unencumbered (bare head) scan with anatomical landmarks.	cload 101_53p 101_53p.lnd
12. Register (align) this scan with the encumbered scan. In effect, this aligns the subject with the helmet scan.	lregister 5 4
13. Copy the pupil landmark locations to the helmet scan. This associates the pupil locations with respect to the global helmet system. Then hide the subject scans.	1 copyland l3 5 l34 copyland l4 5 l38 hide 4 hide 5
14. Load the third subject's encumbered (helmeted) scan with landmark files containing both helmet reference landmarks and anatomical landmarks.	cload 102_53ph 102_53ph.lnd
15. Register (align) this scan with the helmet scan using the common helmet landmarks.	zregister 6 1
16. Load the third subject's unencumbered (bare head) scan with anatomical landmarks.	cload 102_53p 102_53p.lnd

Steps	Commands
17. Register (align) this scan with the encumbered scan. In effect, this aligns the subject with the helmet scan.	lregister 7 6
18. Copy the pupil landmark locations to the helmet scan. This associates the pupil locations with respect to the global helmet system.	1 copyland l5 7 l34 copyland l6 7 l38 hide 6 hide 7
19. Load the fourth subject's encumbered (helmeted) scan with landmark files containing both helmet reference landmarks and anatomical landmarks.	cload 104_53ph 104_53ph.lnd
20. Register (align) this scan with the helmet scan using the common helmet landmarks.	zregister 8 1
21. Load the fourth subject's unencumbered (bare head) scan with anatomical landmarks.	cload 104_53p 104_53p.lnd
22. Register (align) this scan with the encumbered scan. In effect, this aligns the subject with the helmet scan.	lregister 9 8
23. Copy the pupil landmark locations to the helmet scan. This associates the pupil locations with respect to the global helmet system.	1 copyland l7 9 l34 copyland l8 9 l38 hide 8 hide 9
24. Load the fifth subject's encumbered (helmeted) scan with landmark files containing both helmet reference landmarks and anatomical landmarks.	cload 105_53ph 105_53ph.lnd
25. Register (align) this scan with the helmet scan using the common helmet landmarks.	zregister 10 1
26. Load the fifth subject's unencumbered (bare head) scan with anatomical landmarks.	cload 105_53p 105_53p.lnd
27. Register (align) this scan with the encumbered scan. In effect, this aligns the subject with the helmet scan.	lregister 11 10

Steps	Commands
28. Copy the pupil landmark locations to the helmet scan. This associates the pupil locations with respect to the global helmet system.	1 copyland l9 11 l34 copyland l10 11 l38 hide 10 hide 11
29. Convert to alternate landmark representation and view the pupil envelopes for the helmet from different viewpoints.	alt_land right left back front

3.5: Tutorial Five: Radial Difference Maps and Script Parameters

This tutorial demonstrates how a combination of INTEGRATE commands can be used to quantitatively evaluate the radial differences between cylindrical surface scans. It also shows the use of script parameters to create general-purpose, reusable scripts that can be applied in a variety of contexts. Given two scans, the differences can be calculated along each radial value from a reference scan to a second scan. This is referred to as a Radial Difference Map (RDM). For this example, the chest of a subject scanned with a Vitronic Vitus scanner is compared to the chest of the same subject scanned with a Cyberware WB4 scanner. A radial difference map indicates the degree of difference between the 2 scans for this subject. Figure 9 is an RDM of the differences between the two scans. Minor postural differences appear to be the dominant source of differences between the scans.

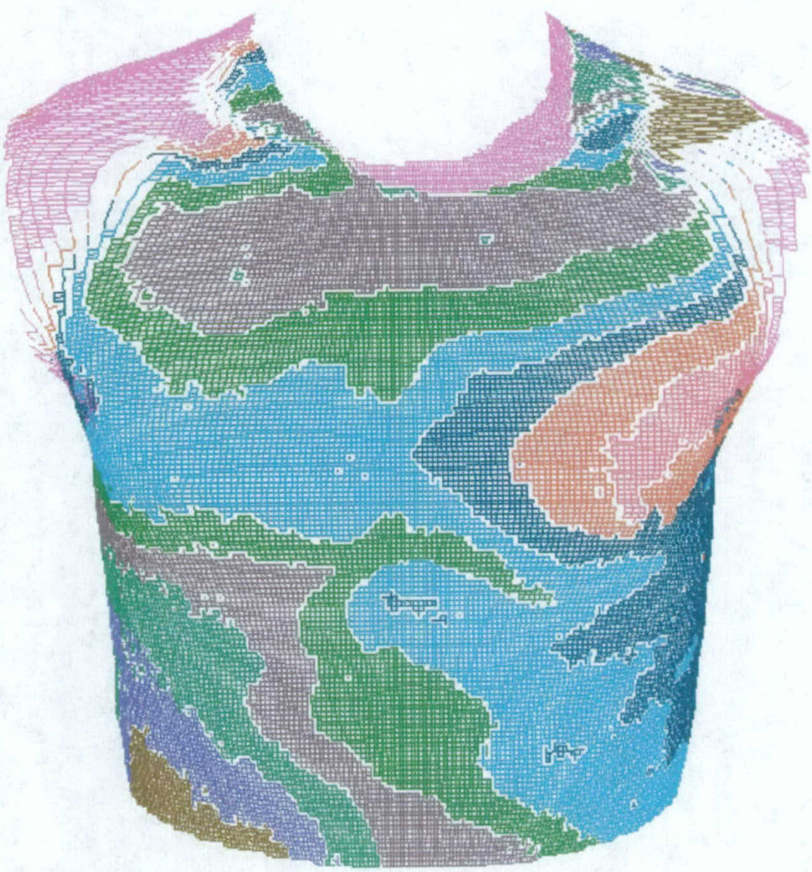


Figure 9: Radial Difference Map (RDM) of the chest region from scans of the same subject using different body scanners. Contrasting colors (monochrome for this publication) represent different degrees of fit

NOTE: For this example, the two scans have been registered to align the surfaces, resampled to transform both into the new coordinate system, and trimmed to the same values. All of these steps are required before performing a radial difference map.

The figure above actually shows 4 threshold levels but the tutorial has been shortened to use only 2 thresholds. The tutorial may easily be extended to 4 threshold levels as an exercise.

The following files are needed for this tutorial:

comp1a1.thorax.cdd, tno1a1.thorax.cdd

An example use of the script shown here might be: **@tutorial_5.txt 10 20**. When working through this tutorial manually, replace all instances of **%1** with **10** and **%2** with **20**.

Steps	Commands
1. Enter INTEGRATE and begin session.	integrate
2. Load the scan file containing the reference chest region.	cload comp1a1.thorax.cdd
3. Move the scan to the center of the screen.	move 0 200
4. Load the scan file containing the chest region to compare, and move it to the center of the screen.	cload tno1a1.thorax.cdd move 0 200
5. Make several copies of the object to be compared. Objects 2-7 will be modified to show different distance ranges between the two original objects.	copy copy copy copy copy
6. Perform a positive subtraction of the two thoraces. The second object will be replaced by the subtraction results. Negative values (object 2 radii greater than matching object 1 radii) will be set to 0.	possub 1 2
7. Eliminate all radial values of the compare object found to be greater than 10 mm above those of the reference object. **Remember that %1 is a parameter presumed set at 10 for this example. Use 10 instead of %1 when performing this tutorial manually.**	threshold 2 ge %1

Steps	Commands
8. Save those radial values in the reference scan where the radial difference is greater than 0 and less than +10 (%1) mm. in the second (replace) object. Save the results into a file called "0ToPlus10".	and 1 2 2 cwrite 0ToPlus%1
9. Perform a negative subtraction of the third object (another copy of the replace object) with respect to the first object. The third object will be replaced by the subtraction results. Negsub sets all positive differences to 0 and sets all negative differences to the absolute value of the difference.	negsub 1 3
10. Eliminate all radial values of the compare object found to be more than 10 (per parameter %1) mm below those of the reference object.	threshold 3 ge %1
11. Identify these radial values for the reference scan. The areas of the reference scan where the radial difference is less than 0 and greater than -10 will be saved as the third object. Save to a file as above.	and 1 3 3 cwrite Minus%1To0
12. Perform a positive subtraction of the compare object with respect to the reference object. The fourth object will be replaced by the subtraction results.	possub 1 4
13. Keep the radial values of the replace object found to be more than 10 mm (%1) greater and less than 20mm (%2) greater than those of the reference object.	threshold 4 lt %1 threshold 4 ge %2
14. Save the radial values for the reference scan where the radial difference is between +10 and +20 in the fourth object. Save to a file as above.	and 1 4 4 cwrite Plus%1ToPlus%2
15. Perform a negative subtraction of the replace object with respect to the reference object. The fifth object will be replaced by the subtraction results.	negsub 1 5
16. For the radial values of the replace object found to be less than those of the reference object, eliminate all difference values less than 10 mm and greater than 20 mm.	threshold 5 lt %1 threshold 5 ge %2
17. Save the radial values for the reference scan where the radial difference is between -10 and -20 mm in object 5. Save to a file as above.	and 1 5 5 cwrite Minus%2ToMinus%1

Steps	Commands
18. Perform a positive subtraction between the reference object and the copy of the replace object stored in object 6.	possub 1 6
19. Remove all radial differences below +20 (%2) mm to 0.	threshold 6 lt %2
20. Save the radial values for the reference scan where the radial difference is greater than +20 mm in object 6. Save to a file as above.	and 1 6 6 cwrite abovePlus%2
21. Perform a negative subtraction between the reference object and the copy of the replace object stored in object 7.	negsub 1 7
22. Set all radial differences below -20 mm to 0.	threshold 7 lt %2
23. Save the radial values for the reference scan where the radial difference is less than -20 mm in object 7. Save to a file as above.	and 1 7 7 cwrite belowMinus%2
24. Hide the reference object so the difference objects can be seen.	hide 1

3.6: Tutorial Six: Helmet Clearance

This tutorial demonstrates an analysis of the fit of a PASGT helmet as actually worn by a subject. The fit criterion is that the inner surface of the helmet should be at least 12.5 mm from the head at all points. To test this criterion, first a helmeted scan is aligned with an unhelmeted scan, then the helmet alone is aligned with the helmeted scan. This brings the bare head and the helmet into the same relationship observed when the subject was wearing the helmet.

To approximate the inner surface of the helmet, we shrink a copy of the helmet outer surface 9.4 mm, which is the reported thickness of the PASGT helmet. This is not exact, since, among other things, the rivets that bulge on the outside of the helmet would also bulge inward on the inside of the helmet, which is not true of our approximated inner surface, but the analysis procedures remain the same when a more accurate inner surface model is available.

Note also the surface of the head is approximated by a cap placed over the hair to compress the hair and improve the visibility of the laser beam during scanning. This is also an approximation, and to be more exact, the depth of the cap and compressed hair should be determined and subtracted from the scanned head surface before checking the clearance.

After bringing the approximated helmet inner surface into the observed relationship with the approximated outer head surface, we begin our clearance check.

After the clearance check is complete, we have 2 products: a new object which represents all points on the head closer than the specified clearance criterion, and a histogram and list of the closest point on the helmet for each point on the head. The surface area of the new object can be determined as a measure of the fit of the helmet on the subject. Similarly, the histogram can be plotted to analyze the clearance distances between the helmet and head.

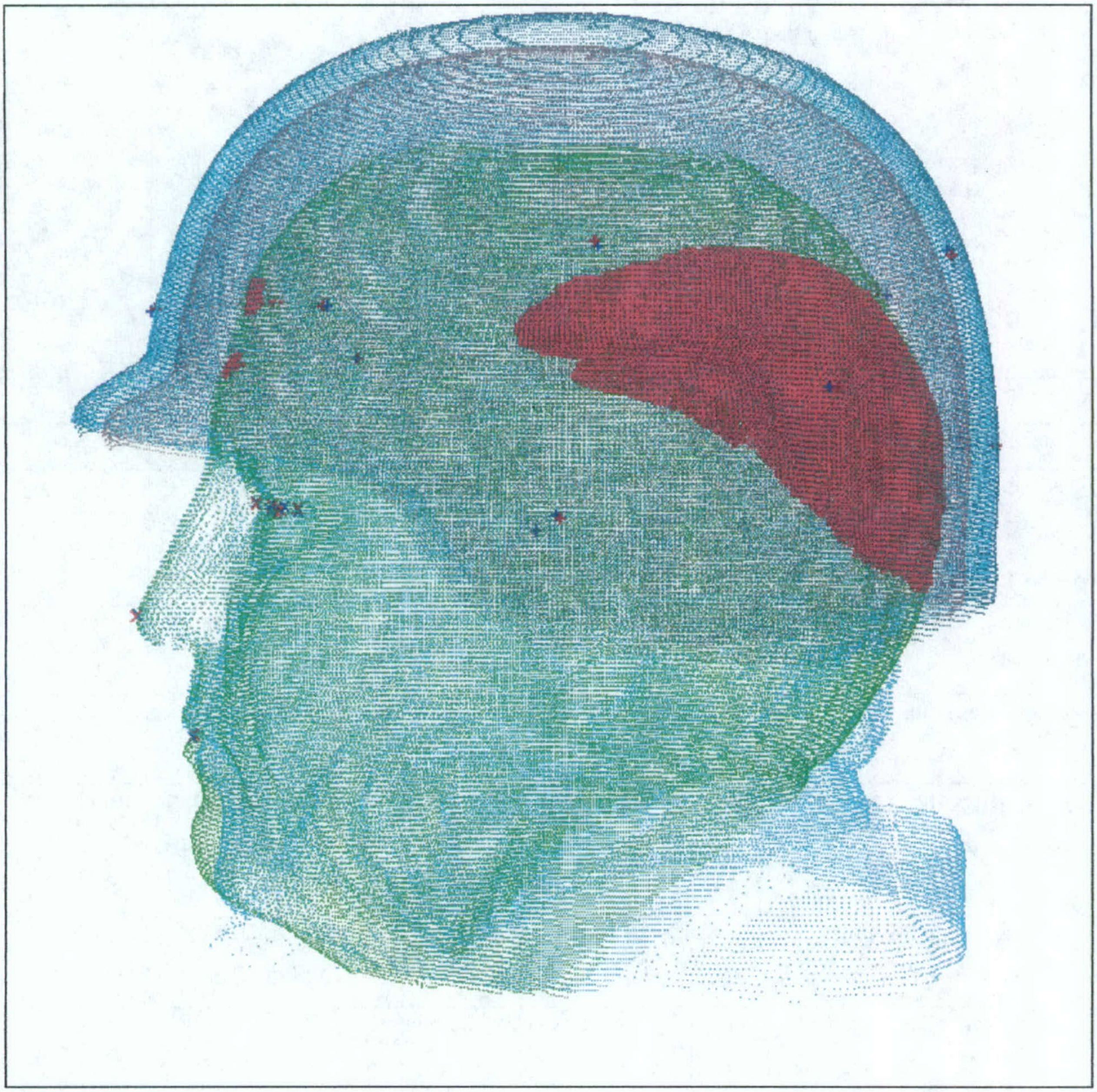


Figure 10: Picture of the helmet clearance with respect to the subject's head scan

In this example, the landmarks have been identified and saved to a landmark (*.lnd) file for each image file. "head" files are of the unencumbered scans, "pasgt" files are of the helmet datasets, and "head.pasgt" are of the head in the helmet.

The following files are needed for this tutorial:

head.g, head.lnd

pasgt.g, pasgt.lnd

head.pasgt.cdd, head.pasgt.lnd

Files produced:

head.results (list of closest points and histogram of closest distances)

head.pasgt.fail.g (dataset with all head points failing clearance criterion)

Steps	**Commands**
1. Enter INTEGRATE and begin session.	integrate
2. Set the viewpoint 1200 mm from the origin along the Z axis. This is equivalent to "front 500" when the eye distance is set at the default distance of 700 mm.	eye 0 0 1200
3. Load unencumbered head scan and landmarks.	gload head.g lload head.lnd
4. Load outer helmet surface and landmarks.	gload pasgt.g lload pasgt.lnd
5. Load helmet head scan and landmarks and hide this image.	cload head.pasgt.cdd lload head.pasgt.lnd hide
6. Register the helmeted scan to the head scan and then register the helmet with the helmeted scan.	lregister 3 1 zregister 2 3
7. Approximate the inner helmet surface by creating another helmet object (movie_seg) and shrinking it by the nominal 9.4 mm thickness of the helmet (shrink). This creates object 4.	2 movie_seg - land off shrink 9.4 0 0 0
8. Remove irrelevant points from head scan to speed things up and create another head object (object 5).	1 movie_seg ly110 hide 1

Steps	Commands
9. Create object 6 which contains the area of the head (object 5) within 12.5 mm of inner helmet surface (object 4) with a 0.5 mm histogram bin size, histogram and intermediate results stored in a file named **head.results**. This operation will take several minutes (depending on the speed of the computer you are using), but it will finish; please be patient. (**NOTE:** This may not be complete due to missing data at the top of the head. You may use the **toupee** command to approximate the surface at the top of the head for more complete results.)	clearance 4 .5 12.5 head.results
10. Store the object containing all points failing clearance criterion.	gwrite head.pasgt.fail.g

3.7: Tutorial Seven: Calipers/Tape Measure

This tutorial demonstrates the use of virtual calipers and tape measure on the whole body data set. For this example, a whole body scan is loaded into INTEGRATE and the calipers are used to measure chest depth and the tape measure is used to record waist circumference at omphalion. Shown in Figure 11 is the whole body data with the calipers.

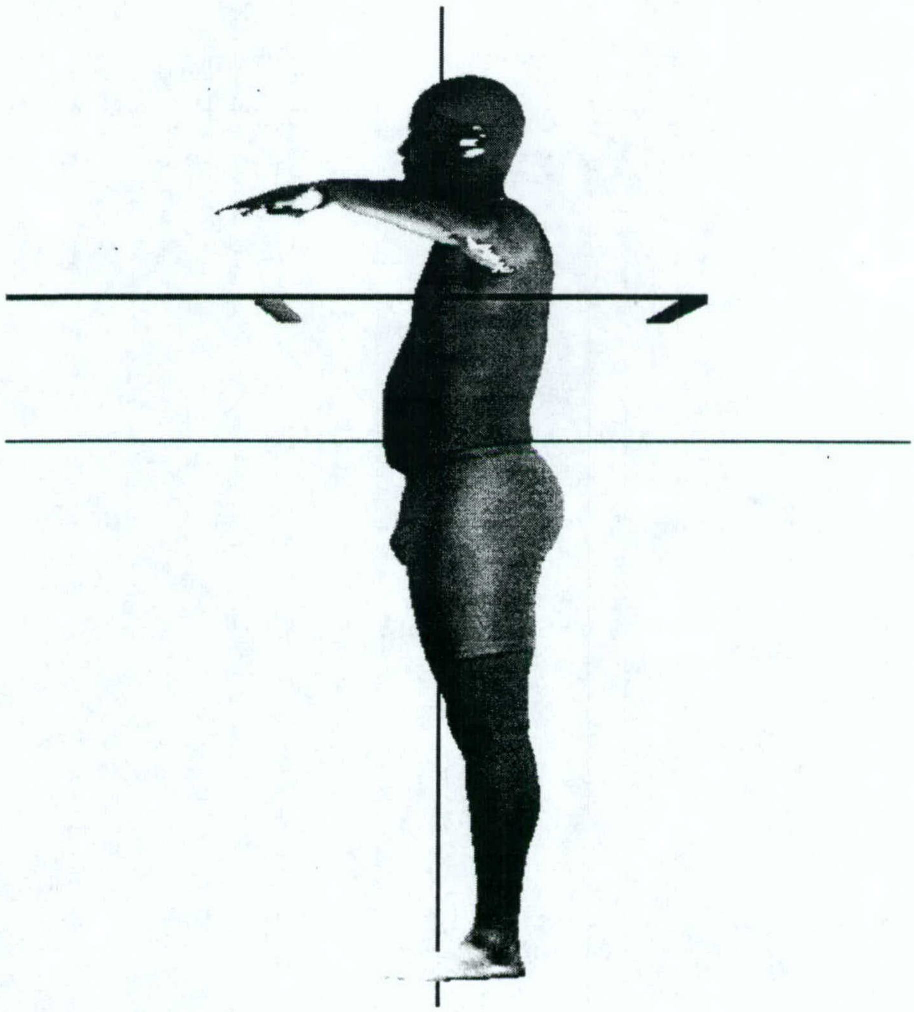


Figure 11: Full body scan data with the virtual calipers

The files needed for this tutorial are:

tsa_stda.ply, std2.mtx

c50x25.g

calipers.g.color, caliper.mtx, waist_circ.mtx

Steps	**Commands**
1. Enter INTEGRATE and begin session.	integrate
2. Establish general setup options.	rgb on gouraud on option wireframe off option points off option surface on
3. Load and position a whole body data set. **(The -1000 resolves an ambiguity with some early dataset scaling and is ignored for newer datasets.)**	pload tsa_stda.ply -1000 mload std2.mtx
4. Load a caliper object (50 cm bar and 25 cm jaws) and enable special caliper processing.	gload c50x25.g fullcolor calipers.g calipers on
5. Orient the calipers for chest depth measurement on the whole body scan.	mload caliper.mtx jaw 2 150
6. Automatically position caliper jaws to "touch" surface (within 0.001 mm (default)) without penetrating the surface at any point.	top auto_jaws 1
7. Set walls to show calipers more clearly.	walls 640 643

Steps	Commands
8. Establish cyberware object to generate circumference measurement. **Movie_seg** is used to segment the torso from the whole body and **resample** is used to create a cyberware object which is compatible with commands such as **cir3p**. Note: the segmented torso object must be translated such that the y axis is centered in the middle of the torso. This is to ensure that the resampled object has a uniform resolution.	walls full right movie_seg uy40 ly-20 hide 1 hide 2 top mload waist_circ.mtx resample hide 3
9. The function to pick points is now turned on and pickmode **cir3p** is selected to establish a horizontal cut plane through the waist to establish a contour representative of waist circumference.	pick on pickmode cir3p
10. Use the mouse to put the cursor at the location of the first (of three) landmarks to be picked. These are points, in this case, selected by the user to represent the level at which the circumference will be taken. Space the three landmarks radially around the abdomen. Look in the global status window for the prompt that lists the next landmark to be picked.	
11. Change the view to see the contour shape.	top surface
12. Determine the distance of the exact contour. Note: "2" in this case is the number associated with the contour. This number may change and will be visible with the contour. (Click on the information box after the measurement is displayed.)	distance 2
13. Now determine the distance of the contour with the virtual tape measure. Notice the measurement is slightly smaller and more representative of an actual tape measure.	tape 2 distance 2

3.8: Tutorial Eight: Establishing Joint Centers on Whole Body Data

This tutorial demonstrates the ability to create landmarks representing joint centers given a set of anatomical landmarks located on a whole body data set. The joint centers are estimates and are assumed to be derived from landmarks representing bony structures on the body. An example is establishing the knew joint as the midpoint between the medial and lateral femoral condyles. A description of the joint center estimates is found in Appendix H. A landmark list complete with illustrations and descriptions is found in Appendix C. Shown in Figure 12 is the whole body with the estimated joint center locations.

Figure 12: Full body scan data with the estimated joint center locations

The files needed for this tutorial are:

csr0099a.ply (or csr0099a.ply.gz if gunzip is available), csr0099a.lnd, std.mtx

Steps	**Commands**
1. Enter INTEGRATE and begin session.	integrate
2. Load the whole body data set, position it to standard position, and create a copy. (One object is to associate with the anatomical landmarks and the other is to associate with the joint centers.) (To parameterize the script, replace all 0099's with %1.)	pload csr0099a.ply mload std.mtx **(or "rotate 0 0 52.5", "rotate -90" if std.mtx not available)** movie_seg -
3. Load the anatomical landmarks into object 1.	1 lload csr0099a.lnd
4. Establish the approximate head center point.	split L1 Z5 Z7
5. Establish the neck/head joint center.	copyland L2 1 L1 add_to_land L2 0 -30 0
6. Establish the neck/thorax joint center, estimated to be at front of spine 1 inch below cervicale.	copyland L3 1 Z24 add_to_land L3 0 -25 51
7. Establish the thorax/abdomen joint center at front of spine at height of 10^{th} rib midspine.	copyland L4 1 Z25 add_to_land L4 0 0 51
8. Establish the abdomen/pelvis joint center at front of spine at level of PSIS marks.	split L5 Z26 Z27 add_to_land L5 0 0 51
9. Establish the right shoulder joint center 1.5 inches below and 1.5 inches inside of right acromion.	copyland L6 1 Z29 add_to_land L6 38 -38 0
10. Establish the right elbow joint center between epicondyles.	split L7 Z34 Z35
11. Establish the right wrist joint center between styloids.	split L8 Z31 Z39
12. Establish the left shoulder joint center.	copyland L9 1 Z41 add_to_land L9 -38 -38 0
13. Establish the left elbow joint center.	split L10 Z46 Z47
14. Establish the left wrist joint center.	split L11 Z43 Z51

Steps	Commands
15. Establish the right hip joint center. (Due to McConville segmentation, Hip joint is used as pivot for both "flap" and thigh, so duplicated and slightly offset.) **(&1Z21Z-&1L12Z is the Z-axis difference between Z21 (Rt. Troch) and L12, changing Z coord of L12 to match Z coord of Z21.)**	split L12 Z17 Z73 add_to_land L12 0 0 (&1Z21Z-&1L12Z) copyland L13 1 L12 add_to_land L13 0 -15 0
16. Establish the right knee joint center between epicondyles.	split L14 Z54 Z55
17. Establish the right ankle joint center between lateral malleolus and sphyrion.	split L15 Z57 Z59
18. Establish the left hip joint center. (The "magic" in the 2nd line moves the landmark forward or backward to the anterior/posterior coordinate of the trochanterion. When doing the tutorial manually, use "land on" (F5) to display the landmark coordinates, then use the difference between the Z23 Z-coordinate and the L16 Z-coordinate.)	split L16 Z19 Z73 add_to_land L16 0 0 (&1Z23Z-&1L16Z) copyland L17 1 L16 add_to_land L17 0 -15 0
19. Establish the left knee joint center.	split L18 Z64 Z65
20. Establish the left ankle joint center.	split L19 Z67 Z69
21. Copy joint centers to aux landmarks in other object for saving.	2 copyland Z1 1 L1 19

Steps	Commands
22. Give joints useful names before saving.	nameland Z1 HeadCenter nameland Z2 HeadNeckJoint nameland Z3 NeckThoraxJoint nameland Z4 ThoraxAbdomenJoint nameland Z5 AbdomenPelvisJoint nameland Z6 RightShoulderJoint nameland Z7 RightElbowJoint nameland Z8 RightWristJoint nameland Z9 LeftShoulderJoint nameland Z10 LeftElbowJoint nameland Z11 LeftWristJoint nameland Z12 RightUpperHipJoint nameland Z13 RightHipJoint nameland Z14 RightKneeJoint nameland Z15 RightAnkleJoint nameland Z16 LeftUpperHipJoint nameland Z17 LeftHipJoint nameland Z18 LeftKneeJoint nameland Z19 LeftAnkleJoint
23. Hide object 1 and show object 2 with the estimated joint centers.	hide 1 points off surface on
24 Save joint landmarks for later use.	lwrite csr0099a.joints.lnd

3.9: Tutorial Nine: Segmenting Whole Body Data

This tutorial demonstrates visualization and manipulation of whole body scan data. The format of the whole body image is considerably different from head scan data and sometimes requires different commands for manipulating this object. An example of this is "eyepoint". To view the entire image, the eyepoint is changed from the default value of 700 mm to about 3000 mm. This allows the user to visualize the whole body data within the bounds of the screen axis system. Commands to segment the whole body data are also demonstrated. Complete segmentation scripts are provided as an attachment, but only the head and neck are demonstrated in this tutorial. See Figure 13.

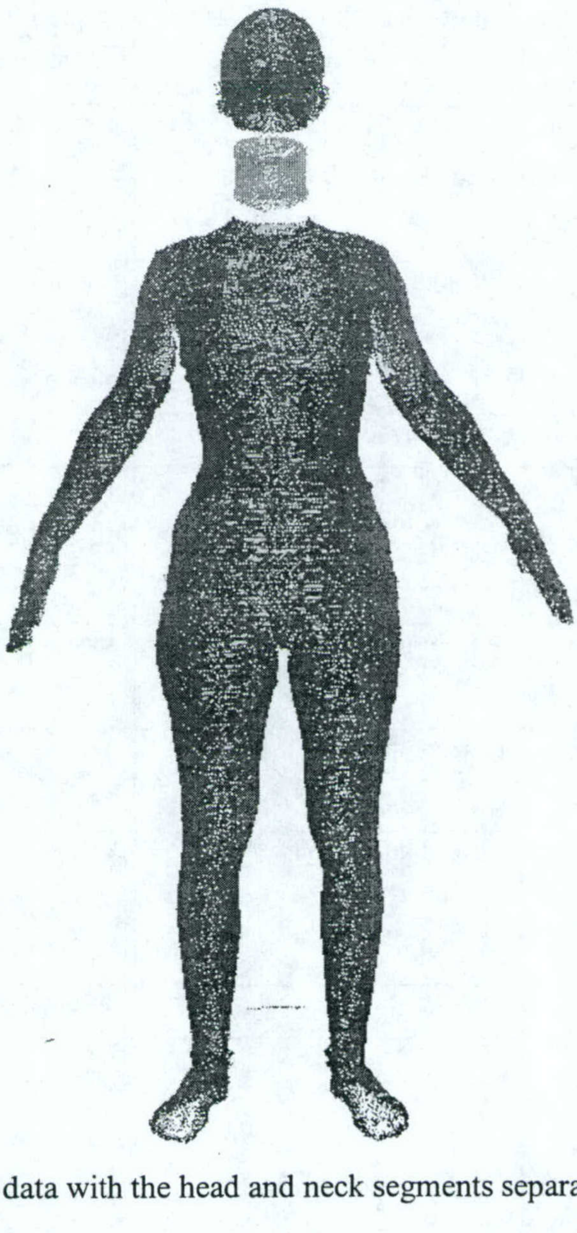


Figure 13: Full body scan data with the head and neck segments separated from the body and limbs

The files needed for this tutorial are:

csr0099a.ply (or csr0099a.ply.gz if gunzip is available), csr0099a.lnd

Steps	**Commands**
1. Enter INTEGRATE and begin session.	integrate
2. Load subject and landmarks into INTEGRATE, then position in "standard" (+Z front) position. **(Note: When doing this tutorial manually, substitute 0099 for every occurrence of %1.)**	pload csr%1a.ply lload csr%1a.lnd rotate 0 0 52.5 rotate -90
3. Create torso/limbs without head/neck. This takes 2 steps, due to the compound plane defining the boundary. First a horizontal cut is made at Z24 (Cervicale).	center z24 movie_seg uy0
4. Now cut at a 45-degree angle through Z10 (Right Clavicale).	rotate -45 center z10 movie_seg uy0
5. Save body/limbs with head/neck removed.	pwrite csr%1a.bodyLimbs.ply
6. Now isolate head/neck by removing body/limbs points from original scan. **THIS WILL TAKE AWHILE. Note that the "disjoint" command does not create a new object.**	1 disjoint 3
7. Align head/neck segment along head/neck cut plane defined by gonions and nuchale.	align xz z8 z6 z9 z9
8. Isolate head from head/neck segment, and save it.	movie_seg ly0 pwrite csr%1a.head.ply
9. Isolate neck from head/neck segment and save it.	1 movie_seg uy0 pwrite csr%1a.neck.ply
10. Clean up.	delete 1-4

3.10: Tutorial Ten: Articulating/Re-Posing Whole Body Data

This tutorial combines the joint centers from Tutorial 8 and the segmentation from Tutorial 9 to demonstrate articulation or re-posing of whole body data. Many whole-body design problems require a specific posture to fit into a scanned piece of clothing or equipment or into the constraints of a workstation layout. It is clearly impractical to scan every subject in every possible pose, so the ability to re-pose a scan to conform to a specific design problem is frequently needed. This tutorial demonstrates techniques for re-posing scans as necessary to address specific design problems. See Figure 14.

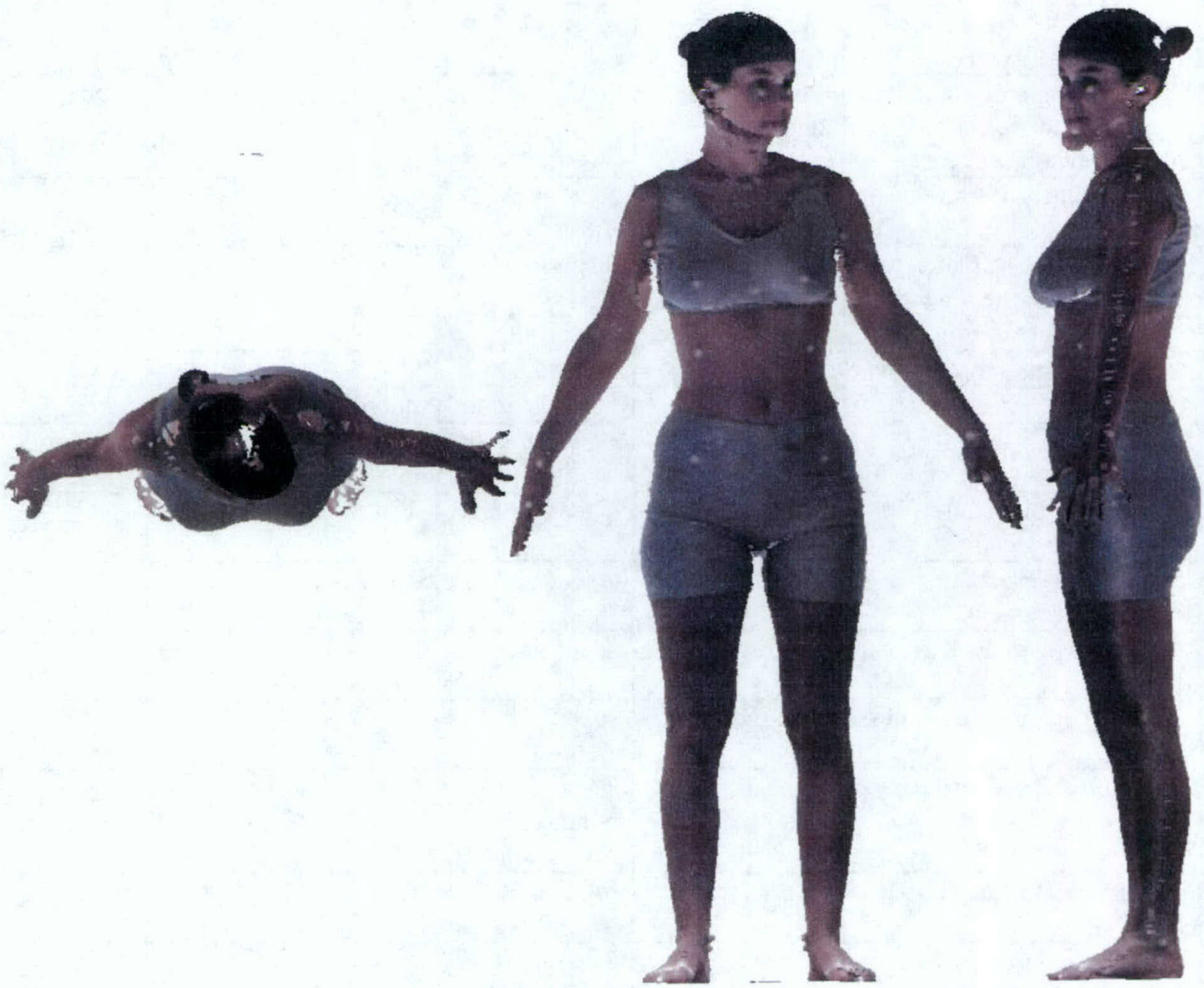


Figure 14: 3 Views Of Articulated/Reposed Subject With Head Turned Left 40 Degrees

The files needed for this tutorial are:

csr0099a.bodyLimbs.ply, csr0099a.head.ply, csr0099a.neck.ply,

csr0099a.joints.lnd (assuming %1 is replaced with 0099), loadLink.txt

Steps	Commands
1. Enter INTEGRATE and begin session.	integrate
2. Open up distance clipping walls.	walls 10 10000
3. Set default eye distance to simplify view commands.	eyedist 1500
4. Apply new walls and new eye distance.	front
5. Load previously segmented body segment.	pload csr%1a.bodyLimbs.ply
6. Load joint center landmarks.	lload csr%1a.joints.lnd
7. Hide joint centers.	land off
8. Move object to "standard" position (+Y up, +Z front)	mload std.mtx
9. Create a temporary landmark at the origin so we can return to the original position after creating a new object center.	newland z20 0 0 0
10. move object so desired center point is at origin.	center z3
11. Make new coordinates permanent, making z3 the new center of the object.	displace
12. Return object to original position.	center z20
13. Remove temporary landmark.	delland z20
	* load and link subordinate objects
	* result is that neck rotates around bodyLimbs and head rotates around neck
14. Invoke sub-procedure file to load and position neck segment.	@loadLink.txt %1 neck 2
15. Invoke sub-procedure file to load and position head segment.	@loadLink.txt %1 head 3
16. Attach head to neck at head-neck joint center.	super link 2 3
17. Attach neck to body/limbs at neck-thorax joint center.	super link 1 2
18. Turn on colored surface view of head.	3 points off surface on
19. Rotate head 20 degrees left with respect to neck.	rotate 0 20
20. Turn on colored surface view of neck.	2 points off surface on
21. Rotate neck 20 degrees left with respect to thorax.	rotate 0 20
22. Turn on colored surface view of torso/limbs.	1 points off surface on

Commands in Procedure File loadLink.txt

Steps	Commands
1. Load and position body segment. **Note parameter 1 is the subject number (e.g. 0099), parameter 2 is the segment name (e.g. head), and parameter 3 is the landmark number of the joint center for the segment.**	pload csr %1a.%2.ply mload std.mtx
2. Get joint center around which this segment rotates from the main/root object.	copyland z1 1 z%3
3. Create a temporary landmark so we can return to the original position.	newland z2 0 0 0
4. Center the segment around its joint center by moving the joint center to the origin of the axis system.	center z1
5. Change the coordinates of all points in the segment so that the origin of axis system is at the joint center.	displace
6. Return the segment to its original position.	center z2
7. Remove temporary landmark z2.	delland z2
8. Hide the joint center landmark.	land off

4.0: INTEGRATE COMMANDS

This command list briefly describes the INTEGRATE commands in alphabetical order. Each description explains the use of the command and the required parts of the command. The Usage, Example, and Result section of each description demonstrates how to set up a command, what an actual command might contain, and what would happen in INTEGRATE if the example command was executed. In the Usage line, parameters that appear in parentheses () are required; parameters that appear in brackets [] are optional.

Executing commands

Many simple INTEGRATE commands can be executed with the function keys. The function key commands appear at the top of the INTEGRATE screen. If the function key list disappears, press function key F7 or type "fkeys" (as always, followed by the <Enter> key) to display it again.

Many simple commands can also be executed by pressing the right mouse key and selecting the command from the menu windows.

Starting with Version 2.8, INTEGRATE supports menu-based file selection through the commands "dir", "open", and "save". When Tcl/Tk is installed on the computer system, these optional commands simplify the process of loading and writing files. The "dir" command is analogous to the "cd" command in setting a preferred input source directory, the "open" command is analogous to the various "load" commands, and the "save" command is analogous to the various "write" commands. Not all file types are supported yet by the open and save commands, but the ones used most frequently are available.

Commands that require additional parameters (for example, the distance to move an object on the screen) must be executed from INTEGRATE's command line. The command line is at the bottom left of the screen, just above the first blue information box, and is marked by an underline cursor. It is possible to configure the function keys for a "partial command". A partial command is copied onto the command line without executing it, allowing the user to modify or add parameters as applicable before pressing the <Enter> key to execute the command.

Toggle commands

Toggle commands, such as wireframe, surface, and land, turn INTEGRATE features on or off, like a light switch. For example, enter **wireframe** to display an object's wireframe and enter **wireframe** again to turn off the wireframe display.

Nobody's Perfect

INTEGRATE is a powerful software tool, but it has some limitations of which the novice user should be aware:

1. There is no "undo" command. Once an INTEGRATE command is entered, it has to run its course. The INTEGRATE operator should double-check each command before executing it. When in doubt, copy the object before performing the operation.
2. INTEGRATE does not discriminate between file types. When the operator mistakenly attempts to load a landmark file with the command for loading an image file, INTEGRATE tries to execute the command. This can produce unexpected results, but it provides great flexibility for naming files.
3. Perspective commands (front, back, and side) sometimes need adjustment. When INTEGRATE loads an image file, the image may be in an awkward orientation. In order to make the front, back, and side commands work correctly, the operator should enter the front command and then use rotate commands to reorient the image. Once the image is oriented correctly for one perspective command, the other perspective commands should work as expected.

LIST OF COMMANDS

<object number>
!
@
$
abssub
add
addobj
add_to_land
align
alt_land
and
auto_jaws
avgland
axes
back
balltest
black
bottom
bottom_cap
boxes
calipers
cd
center
centroid
circumference
clearance
cload
cloads
colors
comment
conclose
contour
contours
copy
copyland
copyseg
cursor
cwrite
cybermovie
delete
delland
delpnt
delseg
delta
derive
diff
dilate
dir
disjoint
displace
distance
do fill
do smooth
drawline
erode
exit
eye
eyedist
fcmod
fcwrite
fill
filter
filtseg
fix_seam
fk
fkeys
force_lnd
front
fullcolor
gcv
gload
gouraud
gwrite
help
hide
histogram
ident
interpolate
intersect
intrplnd
jaw
jump
land
landlist
left
lload
lmlist
lregister
lwrite
man
mark
median
merge
mergeSubs
mirror
mload
modland
move
move_vertex
movie_segment
mwrite
nameland
negsub
newcenter
newland
new_meas
new_order
new_vertex
open
option
ortho
pause
pick
pickmode
planes
pload
points
pop
possub
print
pshrink
push
pwrite
readout
recolor
refresh
remark
resample
rgb
right
rotate
ruin
save
scwrite
select
set
shade
show
show_hid_lnd
shrink
side
skip
skipto
sleep
smooth
split
store
subject
super
surface
surface_area
surf_register
tape
text
thin
threshold
top
toupee
transparent
trim
volume
wait
walls
white
wireframe
wload
wwrite
xload
xwrite
zload
zregister
zwrite

(object number) This is a shortcut for “Select (object number)”. See the Select command for more information.

!(shell command) The ! command prefix activates a shell command from within INTEGRATE, either from the command line or from a script file. It automatically pushes the INTEGRATE window so the user can see other windows to observe actions resulting from the command. After the command is complete, INTEGRATE pops to the surface when the user presses any key.

$(shell command) The $ command prefix activates a shell command from within INTEGRATE, either from the command line or from a script file. It does not change any window configuration, and it does not wait for the user to press any key after the command is completed.

@(filename) The @ command prefix activates a script file specified by (filename). It allows commands to be grouped into standard sequences to reduce mental gymnastics and repetitive typing.

A command file (without the @) can also be specified as part of the command line that starts INTEGRATE. For example: “integrate load52 spinfast” starts INTEGRATE, loads, trims, and rotates subject 52 (load52), then spins the viewer's eye around the object (spinfast).

Command files, also called script files, can be parameterized (e.g. "@spinvar 2" which provides a parameter of 2 to the spinvar command file), and can provide a limited ability to support non-sequential operations such as looping or if-then-else constructs (see the JUMP command).

abssub

This command performs an absolute subtraction on two objects along each radial value. Specify a reference object and a replace object of the same size (that is, with the same trim values), and INTEGRATE subtracts one object from the other. The second object is subtracted from the first object, and the absolute value of the difference is retained. INTEGRATE stores the subtraction result in the replace object. Unless the objects are first registered and resampled, the results of this operation will probably not be the expected results.

Usage: abssub (reference object) (replace object)
Example: abssub 1 2
Result: INTEGRATE subtracts object 2 from object 1 and stores the result in object 2.

add

This command performs an addition on two objects. The user specifies a reference object and a replace object of the same size (that is, with the same trim values), and INTEGRATE adds the objects together. INTEGRATE stores the addition result in the replace object.

Usage: add (reference object) (replace object)
Example: add 1 2
Result: INTEGRATE adds objects 1 and 2 and stores the result in object 2.

addobj

This command replaces a section of the grid of one object with the same grid section of a second object. Addobj needs two parameters: the object to be modified and the object to be added to the specified object.

Usage: addobj (to object) (from object)
Example: addobj 1 2
Result: Object 1 is now a combination of objects 1 and 2.

add_to_land

This command adds an XYZ offset in the screen coordinate system to the specified standard or auxiliary landmark of the active object. The user specifies a landmark number (for example, L2 for a standard landmark or Z2 for an auxiliary landmark) and an offset for the X-axis. The user can also specify offsets for the y and z axes, but those parameters are optional. An example of this function would be to find the mid-point of the tragions (using the split command) while the head is aligned in the Frankfurt Plane axis system, then to add Beier's constant (8.3, 0, 31.2 mm) to convert the landmark to the approximate position of the Center of Gravity of the head.

Usage: add_to_land (Z#/L#) (X) [Y] [Z]

Example: add_to_land z2 0 31.2 8.3

Result: The auxiliary landmark z2 now represents an estimated position for the center of mass of the head.

align This command aligns an object to the screen axis system according to three points on the object. When the alignment is complete, the first specified point will be at the origin, the second point will be on the first specified axis, and the third point will be on the specified plane. If a fourth point is specified, the object will be moved so that the projection of the fourth point lies on the specified axis. Align needs from four to nine parameters:

The first parameter is two or three characters, which may be x, y, or z. The first character of the first parameter is the name of the axis to be defined by the first two points. The second character of the first parameter is the name of the axis perpendicular to the first axis, and on the plane defined by the first axis and the third point. The third character of the first parameter is the name of the axis along which the object will be shifted if a fourth point is specified. If a fourth point is specified but a third character is not, INTEGRATE shifts the object along the axis defined by the first two points.

The second through ninth parameters specify landmarks or longitude/latitude coordinates. Points can be specified by landmark number (L# or Z#), or by the longitude and latitude of the point. Three or four points must be specified. These points define the axis named above. If a fourth point is specified, INTEGRATE moves the object so that the coordinate of the 4th point corresponding to either the 1st specified axis (or the 3rd axis if specified) is zero.

Usage: align axis1axis2[axis3] land1 land2 land3 [land4]
Example: align xy l1 l2 l3
Result: INTEGRATE rotates the object into a coordinate system defined as follows: the X-axis passes through standard landmarks 1 and 2, the Y-axis is perpendicular to the X-axis passing through standard landmark 3, and the Z-axis is defined as the cross product of the X and Y axes. Because a 4th point was not specified, the origin is defined by standard landmark L1.

alt_land [on/off]

This command toggles the landmark point display from L# or Z# form (landmarks individually labelled) to x or + form (landmarks marked but not labelled). The x/+ form reduces screen clutter when landmark labels are not needed. L# or X designate "standard" landmarks (e.g. Tragions or Infraorbitale), while Z# or + designate "auxiliary" landmarks, which are typically defined only for a specific study.

Usage: toggle command
Example: alt_land
Result: Landmarks change from x or + to L# or Z#.

and

This command performs a logical AND operation on two grid objects. Points with a radius of zero in the objects' data are considered binary zeros, while non-zero radii are considered binary ones. The user specifies a reference object and a replace object of the same size. The values of the reference object are stored in the replace object wherever the two objects AND to a binary one.

Usage: and (reference object) (replace object)
Example: and 1 2
Result: Object 2 is replaced by the radial values of object 1 at the non-zero radial locations of object 2.

auto_jaws

This command automatically performs jaw closure on a calipers object (set as the active object) around a test object (object to be measured). It iteratively moves the jaws, then tests for intersection (see the jaw and intersect commands below). It requires that the calipers be positioned for the measurement, with the jaws definitely outside of (not intersecting) the test object. It positions the jaws until they are within "tolerance" of the test object without intersecting. If "tolerance" is not specified, it defaults to .001. The "bounding_box" value is the distance outside the edges of the bounding box for the calipers in which points are still considered for the intersection test. Limiting this test increases the speed of the operation dramatically. If "bounding_box" is not specified, it defaults to 6.0 mm.

Usage: auto_jaws (test_object) [tolerance] [bounding_box]

Example: auto_jaws 2 0.1 4.0

Result: The calipers jaws are closed to within 0.1 mm of the surface of object 2. Only the object 2 points within 4 mm of the bounding box of the calipers are considered for the intersection test.

avgland

This command averages the standard landmark sets from a selected group of objects to produce a new landmark set which represents the centroids of corresponding landmarks. INTEGRATE attaches the new landmark set to the Active Object. The newly defined landmarks can be left as-is, meaning that they stay exactly where the are computed to be, or they can be projected onto the surface of the Active Object.

Avgland requires at least two parameters: (surf/asis) and a list of objects to be included in the average. Note: a single landmark set may be copied by using avgland with only one object.

Usage: avgland (surf/asis) (obj1) [obj2] ... [objectN]

Example: avgland surf 1 2 3 4

Result: INTEGRATE averages the standard landmarks of objects 1, 2, 3, and 4 and projects the averages onto the surface of the active object.

axes [on/off] This command turns the display of the X, Y, and Z axes on or off.

Usage: toggle command

Example: axes

Result: The axes appear or disappear.

back This command moves the user's "eye" to the back of the object(s).

Note: The objects' coordinates do not change. When back is executed, it is as if the viewer moved behind the object to see the back of it. To change an object's coordinates, use move or rotate.

Back has one optional parameter: a distance. If the distance is positive, the viewer's eye will be positioned that much further than the default distance (normally 700; see eyedist) away from the object. If the distance is negative, the viewer's eye will be positioned that much closer to the object.

Usage: back [+/-number of mm]

Example: back 300

Result: The viewer sees the back of the object, 300 mm further away from it than the default eye distance.

balltest

This command evaluates the accuracy of a scanner by computing radii for a scanned calibration ball or cylinder. By comparing the calculated values with the true values and by analyzing the standard deviation and range values, a user can determine whether a calibration correction is needed and its magnitude. Balltest needs one parameter, the latitude to use for the radius compare.

Use **pickmode point** to select a latitude on the scan of the calibration ball. This is the latitude to specify in the balltest command. A latitude at the midpoint between radius steps (if present) is best to use because of potential scanner errors at sharp edges.

Usage: balltest (latitude)

Example: balltest 125

Result: INTEGRATE computes the average radius of a calibration ball or cylinder and displays the dimensions in the lower left corner of the screen. The range and standard deviation should both be small and the dimensions should match the actual radius of the ball within desired tolerances at the specified latitude.

black

This command sets the screen background color to black. Landmark and object points will change colors so that they will show up against the black background.

Usage: black

Example: black

Result: The screen background color turns black.

bottom

This command moves the viewer's eye to the bottom of the object(s).

Note: The objects' coordinates do not change. When bottom is executed, it is as if the viewer moved under the object(s) to see the bottom of them. To change an object's coordinates, use move or rotate.

Bottom has one optional parameter: a distance. If the distance is positive, the viewer's "eye" will be positioned that much further than the default distance (normally 700; see **eyedist**) from the object(s). If the distance is negative, the viewer's eye will be positioned that much closer to the object(s).

Usage: bottom [+/- number of mm]

Example: bottom

Result: The viewer sees the bottom of the object.

bottom_cap

This command fills in the bottom of the active grid object. Note that this command works best when the object is positioned so that the (estimated!) lowest point on the head is centered about the Y axis. This command requires one parameter: the lowest latitude to establish the bottom plane. Check the object coordinates in the blue box on the lower left for the latitude to use for the bottom plane. The latitude must be within the current trim area.

Usage: bottom_cap (latitude to establish the plane)

Example: bottom_cap 50

Result: INTEGRATE fills in the bottom of the head scan trimmed at lower latitude 50

boxes [on/off]

This command turns the status boxes on or off. This can be useful for making snapshot/screen dumps or for increasing the available viewing area of the screen.

Usage: toggle command

Example: boxes

Result: The status boxes at the bottom of the screen appear or disappear.

calipers [on/off]

This command is used to turn calipers mode on and off once a calipers object is loaded. Tutorial_9 demonstrates how calipers are used in conjunction with the caliper objects and commands such as **jaw**, **auto_jaws**, and **readout**.

NOTE: A calipers object is a normal object representing 2 jaws and a slide bar. It also includes some special processing which is only active when calipers mode is on.

Usage: toggle command

Example: calipers

Result: Calipers mode will turn on or off.

cd

This command changes the directory to be used for loading data files. Note that this command does not work the same way the cd command works in an operating system shell. It is not cumulative, but must always specify the complete path desired.

The format of the cd command is "cd [path]" where path is a standard path descriptor, such as "/data/head/hgu55p". The cd command with no parameter sets the path back to the path where INTEGRATE was started.

Usage: cd [path]
Example: cd /headfiles/survey
Result: The directory for reading data files changes to headfiles/survey.

center

This command moves the active object to put the specified point at the center of the axis system. There are two forms of the command: "center x y z," which names the coordinates of the point to be centered, and "center L#/Z#," which names the landmark (L for a standard landmark or Z for an auxiliary landmark) at the point to be centered.

Usage: center (L#/Z#)/(X Y Z)
Example: center z1
The active object moves so that its auxiliary landmark Z1 is at the center of the axis system.

centroid

This command computes the centroid of either the standard landmarks or the auxiliary landmarks and stores the result in the specified landmark location. Centroid requires two parameters: the landmark to hold the result and whether to compute the centroid of the standard landmarks (L or STD) or the auxiliary landmarks (Z or AUX).

Usage: centroid (Z#/L#) (Z/L/aux/std)

Example: centroid z3 aux

Result: INTEGRATE computes the centroid of the auxiliary landmarks and stores the resulting centroid in landmark Z3.

circumference

This command computes a circumference line completely around the object where a plane specified by either three points or by two specified points and the center of the object intersects the surface of the object.

Usage: circum (L#/Z#/lg lt) (L#/Z#/lg lt) [L#/Z#/lg lt]

Example: circum z2 z3

Result: Integrate creates a line around the circumference of the object on the plane defined by the center of the object and auxiliary landmarks 2 and 3.

Circumference accepts its parameters in a variety of ways: any of the two or three specified points can be a longitude/latitude coordinate pair <lon> <lat>, a standard landmark number (L#), or an auxiliary landmark number (Z#).

If a single parameter is specified, INTEGRATE assumes it is either a latitude or a landmark to use as a latitude reference. The resulting circumference line is drawn along the specified latitude from the left trim limit to the right trim limit. The first point (leftmost longitude) is repeated at the end of the circumference to form a complete circle.

A circumference can also be generated by picking two or three points (pickmode cir2p or cir3p) using the mouse in point picking mode.

clearance

clearance - This command determines the clearance distances between every point on a test object and the closest corresponding point on a reference object. It produces a dataset made up of all points closer than a specified threshold, and a text file containing a histogram showing the distribution of clearance distances for the entire test object. The test object and the reference object must be mesh objects, and the resulting object is also a mesh object.

Clearance takes the following parameters:
ref_obj - the object which is tested against the active object to determine clearances
bin_size - the resolution of the clearance histogram
thresh - a distance below which a point is considered "too close". Points that are too close will be copied to a new object for further analysis. If thresh is not specified, no points are copied.
result - the name of a file in which to store the detail and histogram information. If result is not specified, no histogram is created.

Note that because each point is compared to each point on the other object, the clearance command applied to two large objects will take many minutes to complete.

Usage: clearance ref_obj bin_size [thresh [result]]
Example: clearance 1 0.5 12.5 clearance.results
Result: The active object is compared to object 1. A histogram of clearance distances is produced in file clearance.results at 0.5 mm resolution. All points closer than 12.5 mm will be copied to a new object for separate analysis.

cload, cloads*

This command reads in a scan file. Cload has four optional parameters, one necessary file name, and one optional file name. The parameters are:

a, for an ASCII header;

b, for a binary header;

c, if there is a color file associated with the scan file and the color file is to be read in; and

n, if there is no color file or if you don't want to load the color file.

If any of these parameters are used, group them together and precede the first parameter with a "-" (dash). The current default is binary and no color (-bn).

After the parameters, if any, type the name of the scan file to be read in. After the scan file name, the user can choose to add the name of the landmark file associated with the scan file.

Usage: cload [-abcn] point_file [land_file]

Example: cload headscan headscan.lnd

Result: INTEGRATE reads in a scan file called headscan and its associated landmark file, headscan.1nd.

***cloads is used for loading in stereophotogrammetry segment data.**

colors

This command manipulates the object color table. The colors command can update the background colors or update the object colors.

Updating the background colors requires the following format:

COLORS 0 background text box

where:

background= the background color,

text= the text color,

box= the background color of the information boxes.

Updating the object colors requires the following format:

COLORS n bfeat bpnts wfeat wpnts

where:

n= the object number to change,

bfeat= the color for features, such as landmarks and contour lines, when the background is black,

bpnts= the color for points and wireframe when the background is black,

wfeat= the color for features, such as landmarks and contour lines, when the background is white,

wpnts= the color for points and wireframe when the background is white.

The available colors are:

black = 0	medium gray = 14
red = 1	bluish red = 15
green = 2	greenish red = 16
yellow = 3	bluish green = 17
blue = 4	reddish green = 18
magenta = 5	greenish blue = 19
cyan = 6	reddish blue = 20
white = 7	light red = 21

dim red = 8	light green = 22
dim green = 9	light yellow = 23
dim yellow = 10	light blue = 24
dim blue = 11	light magenta = 25
dim magenta = 12	light cyan = 26
dim cyan = 13	

comment

This command annotates the session audit trail with a text string. The comment appears in the INTEGRATE session record, stored in the directory from which INTEGRATE was launched.

Usage: comment (string)

Example: comment starting new session

Result: "starting new session" appears in the INTEGRATE session record.

conclose

This command closes a series of point-picked contours using pickmode MUL2P or pickmode MUL2A. After the contour points are picked, conclose creates the contour line between the first and last points picked.

Usage: conclose

Example: conclose

Result: INTEGRATE completes the connected contours by drawing a contour line between the first and last points picked.

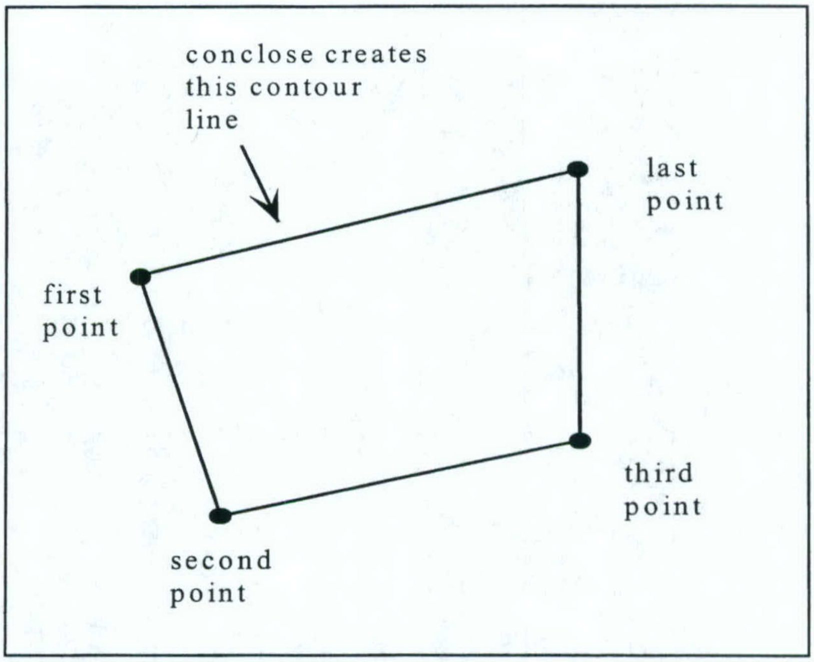


Figure 15: Conclose joining the first and last selected points

contour

This command computes a contour line from one point to another using an optional third point or the object center to establish the plane of the contour.

Usage: contour (L#/Z#/(lg lt)) (L#/Z#/(lg lt)) (L#/Z#/(lg lt))
Example: contour z1 z2
Result: INTEGRATE draws a contour line on the surface of the active object that connects auxiliary landmarks Z1 and Z2 along a plane that passes through the object's center.

Contour accepts its parameters in a variety of ways: any of the two or three specified points can be a longitude/latitude coordinate pair <lon> <lat>, a standard landmark number (L<#>), or an auxiliary landmark number (Z<#>).

If a single parameter is specified, INTEGRATE assumes either a longitude or a landmark to use as a longitude reference. INTEGRATE draws the resulting contour line along the specified longitude from the lower trim limit to the upper trim limit.

A contour can also be generated by picking two or three points (**pickmode** con2p or con3p) using the mouse in point picking mode. Draw a connected series of two-point contours consecutively by using pickmode mul2p, then picking points. With this pickmode, the endpoint of one contour automatically becomes the startpoint of the next contour. Pickmode mul2a also creates a series of two-point contours, but instead of using the object center as in mul2p, it uses the Z-axis at the average latitude of the two points for its third point.

contours [on/off] This command turns the display of contour lines on or off.

Usage: toggle command

Example: contours

Result: Contour lines appear or disappear.

copy This command copies one object to another, eliminating the points trimmed from the first object and keeping only the points designated by the first object's thin factor.

Copy accepts one optional parameter: the number of the object to be copied. If no object number is specified, the active object is copied. Copy creates a new object in the first available object pool slot.

Usage: copy (object)

Example: copy 3

Result: INTEGRATE creates a copy of object 3.

copyland

This command copies one or more landmarks from one object to another. Copyland can consolidate landmarks from two or more sets into a single set. An example might be the consolidation of the Tragions from a subject data set with the landmarks from a helmet scan to allow analysis of the relationship of Tragions to helmet position.

Usage: copyland (Z#/L#) (from object) (Z#/L#) [count]
Example: copyland l1 2 l2 3
Result: INTEGRATE copies landmarks 2, 3, and 4 from object 2 and stores them in landmarks 1, 2 and 3 on the active object.

Copyland operates on the active object and requires three parameters: the destination landmark number (Z# or L#), the source object number, and the source landmark number (Z# or L#). An optional 4th parameter specifies the number of consecutive landmarks to copy. If the 4th parameter is not specified, INTEGRATE only copies the source landmark.

copyseg

This command copies the area of an object bounded by specified contours to a new object. Copyseg needs a list of contours which describe the segment to be copied. Each contour displayed on an object is numbered at approximately the midpoint of the contour. Use this number in specifying contours for copyseg. Two special contours, t and b, are used to specify that the top of the object or the bottom of the object is to be used as a contour boundary. "Copyseg t b" would copy the entire active object (as thinned and trimmed). Copyseg creates a new object in the first available object pool slot.

Usage: copyseg (contour1) ... [contourN]

Example: copyseg 1 2 3

Result: INTEGRATE makes a copy of the segment bounded by contours 1, 2, and 3.

cursor [on/off]

This command enables or disables a surface tracking cursor for the active object. The surface tracking cursor is a crosshair that conforms to the contours of the surface of the object.

Note: When cursor is executed, INTEGRATE redraws the screen each time the cursor moves. To minimize redraw delays, use the cursor command with point display, trim the object to the minimum size needed, and hide as many objects as possible before enabling the cursor.

Usage: toggle command

Example: cursor

Result: The surface tracking cursor appears or disappears.

cwrite

This command writes out a (new) Cyberware-format file. Cwrite requires a new filename as a parameter. It always writes the data with the new/modified ASCII header.

Usage: cwrite filename

Example: cwrite face_scan.cdd

Result: INTEGRATE writes the data in the active object to a new file called face_scan.cdd. INTEGRATE stores the file in the directory from which INTEGRATE was started.

cybermovie

This command copies a rectangular grid (e.g. cyberware echo) object into a new object, converting it to a triangular mesh (e.g. movie.byu/.ply) representation. Cybermovie works on the active object only, and requires one parameter: whether the object is completely closed (WRAP) or is a partial surface (NOWRAP). This parameter determines whether INTEGRATE creates polygons to connect the object's last longitude to the first longitude.

Usage: cybermovie (wrap/nowrap)

Example: cybermovie wrap

Result: INTEGRATE creates a copy of the active (grid) object in mesh format.

delete

This command removes one or more objects from the object pool or one or more sub-objects from an object. Delete has several optional parameters. Parameter 1 is either the number of the object to be removed or a range of objects to removed (e.g. DELETE 1-10). If no parameters are specified, the active object is deleted. After objects have been removed, new objects can be read in to replace them.

If two or more parameters are specified, the first is the object (or objects) to be modified, and the subsequent parameters are the sub-objects to be deleted from the specified object. The object itself is not removed when two or more parameters are specified.

Usage: delete [object]
Example: delete 3
Result: INTEGRATE removes object 3 from the object pool.

delland

This command deletes a landmark value from the standard landmark list. The slot in the list remains, but the coordinates of the landmark are zeroed. Delland requires one parameter, the standard or auxiliary landmark number of the landmark to delete (L# or Z#).

Usage: delland (Z#/L#)
Example: delland z13
Result: INTEGRATE deletes the landmark value stored in auxiliary landmark Z13.

delpnt

This command deletes (voids) one point from the data set. Delpnt accepts its parameter(s) in a variety of ways: the point can be specified as a longitude/latitude coordinate pair <lon> <lat>, a standard landmark number (L#), or an auxiliary landmark number (Z#). Points can also be deleted by picking points with the mouse when pickmode is set to delpnt or delall.

Usage: delpnt (Z#/L#) or delpnt (lon lat)
Example: delpnt z24 or delpnt 189 56
Result: INTEGRATE deletes the point at auxiliary landmark Z24 or the point at longitude 189, latitude 56.

delseg

This command deletes (voids) all of the points within the boundaries of a specified set of contours on the active object. It only works on grid-type objects. Delseg needs a list of contours which describe the segment to be deleted. Each contour displayed on an object is numbered at approximately the midpoint of the contour. Use this number in specifying contours for delseg. Two special contours, t and b, are used to specify that the top of the object or the bottom of the object is to be used as a contour boundary. "Delseg t b" would delete all of the points in the entire active object (as thinned and trimmed).

Usage: delseg contour1... contour#
Example: delseg 2 3 4 5 6
Result: INTEGRATE deletes the region bounded by contours 2 through 6 from the active object.

delta

This command colors the surface of an object according to its distance from a reference object. Points on the object with larger radii are shown in yellow, fading to red with increasing distance. Points on the object with smaller radii are shown in cyan, fading to blue with increasing distance. Delta needs one parameter: the number of the reference object. It always operates on the active object. If no reference object is specified, the distance for color-coding is computed from the mean radius of the active object. If the active object has full color information and RGB mode is enabled, the new colors may not be visible. To view the delta colors, disable RGB mode or delete the full color information. Use the shade command to restore the gray-scale surface lighting colors.

Usage: delta [reference_obj]

Example: delta 3

Result: The color of the active object changes at each point to reflect that point's distance from the equivalent point on object 3.

derive

This command derives a missing landmark from a second landmark and the resultant from combining the two landmarks. Derive needs three parameters and has two optional parameters. The required parameters are a destination landmark number (Z# or L#), the resultant landmark number (Z# or L#), and the other landmark used to produce the resultant (Z# or L#). You can specify an optional total weight and second landmark weight if the resultant was produced with unequal weights.

Suppose you want to create a new landmark, Z3, between landmarks L1 and L6.

• L1 • Z3 • L6

You would use **split** to create Z3, the resultant landmark. If data is lost and L6 disappears

• L1 • Z3

you can recreate it with **derive**. To recreate L6, type

derive L6 Z3 L1

If you used weighting in the split operation, you should use the same weighting in the derive command.

Usage: derive (Z/L destination landmark#) (Z/L landmark 1) (Z/L landmark 2) [weight 1] [weight 2]

Example: derive z5 L28 L14 5 .5

Result: INTEGRATE recreates landmark Z5, using 5 as a total weight and 0.5 as a second landmark weight.

diff This command displays and prints the difference between the rotation angles of two scans. The displayed difference is the difference to rotate one scan into the axis system of the other scan. This command ignores any offset differences due to different centers of rotation. It requires two parameters: the "final" object and the "beginning" object. For example, DIFF 1 2 provides the angles necessary to rotate object 2 into the same orientation as object 1. The display and print show the angular rotation around the X axis, the Y axis, and the Z axis. If then difference in rotation around the Z axis is significant, the other two angles may be in error.

Usage: diff obj1 obj2
Example: diff 1 2
Result: INTEGRATE displays and prints the angles needed to rotate object 2 into the same orientation as object 1.

dilate This command performs a morphological dilation on an object. After an object has been **eroded**, dilate completes the smoothing process. Dilate expands the data so that a structuring element with an origin placed at the original data fits. The structuring element used in this instance is a cylinder with a spherical top. The user specifies the radius of the sphere or both the cylinder and the sphere. (If only the sphere is specified, the cylinder is set to the same radius.) The default dilation operation is positive. A negative dilation can be performed by adding the modifier "minus."

Usage: dilate (sphere size) [cylinder size] [MINUS]
Example: dilate 5 2
Result: If erode has already been executed, INTEGRATE smoothes the active object.

dir

When Tcl/Tk is installed on a system (see INTEGRATE installation instructions), this command provides a file dialog box to allow setting the default input directory graphically. It performs the same function as the "cd" command.

Usage: dir

Example: dir

Result : a dialog box appears that allows the user to select the desired input directory.

disjoint

This command is used in the process of separating a dataset into two or more disjoint segments. One or more segments are extracted from the original segment, typically using movie_seg, then the extracted segments are removed from the main segment using this command. It requires a list of objects whose points are to be removed from the active object to create disjoint point sets.

Usage: disjoint obj#

Example : disjoint 1 2 3

Result : All points shared between the active object and objects 1, 2, and 3 are removed from the active object.

displace

This command applies the present displacement matrix for the active object (which must be a triangular mesh object) to each point in the object, then resets the displacement matrix to the identity matrix (no rotations or translations). This allows a permanent change of axis system when the object is written out (see GWRITE, PWRITE, or WWRITE).

Usage: displace
Example: displace
Result: When the active object is subsequently written out, it will be stored in exactly the orientation as it appears on the screen. When it is re-loaded, it will appear in the same position it was in when the displace/write sequence was executed.

distance

This command computes the total surface distance along either contours or circumferences of the active object. Distance requires a list of contours or circumferences to be measured. Each contour displayed on an object is numbered at approximately the midpoint of the contour. Use this number in specifying contours. Distance computes the contour distance for each contour and displays the sum of all the distances.

Usage: distance (contour1) ... [contourN]
Example: distance 3 5
Result: INTEGRATE displays the surface distance along the contours numbered 3 and 5 on the active object.

do fill This command replaces void points in the active object with an approximation based on surrounding points.

Usage: do fill

Example: do fill

Result: INTEGRATE fills in missing points on the active object (which must be a grid object).

do smooth This command replaces each point in the active object's data set with the average value of the point and its neighbors, resulting in smoother data surfaces. This is a very crude smoothing process and should be used carefully.

Usage: do smooth

Example: do smooth

Result: INTEGRATE smoothes the surface of the active object (which must be a grid object).

drawline This command draws a straight line from one landmark through another landmark, with an optional length specified. Drawline requires two parameters: the landmark at the origin of the line (Z# or L#), and a landmark that the line is to pass through. An optional length parameter specifies the length of the line. If length is not specified, the line ends at the 2nd landmark.

Usage: drawline (L#/Z#) (L#/Z#) [length]

Example: drawline L1 L32

Result: INTEGRATE draws a line connecting standard landmarks 1 and 32.

erode

This command performs a morphological erosion on an object. Erosion shrinks the data so that the origin of a structuring element fitted within the data becomes the new location for a data point. The structuring element used in this instance is a cylinder with a spherical top (see figure 16 below). The user specifies the radius of the sphere or both the cylinder and the sphere. If only the sphere is specified, the cylinder is set to the same radius. The larger the sphere and cylinder, the greater the erosion that occurs. The default erosion operation is positive. A negative erosion can be performed by adding the modifier "minus". (A morphological opening is an erode followed by a dilate. Note that using either erode or dilate alone, or with different parameters, will produce undesirable distortion of the dataset.)

Usage: erode (sphere size) [cylinder size] [minus]

Example: erode 4 5

Result: INTEGRATE takes the "sharp edges" off the data. To complete the smoothing process, execute **dilate 4 5.**

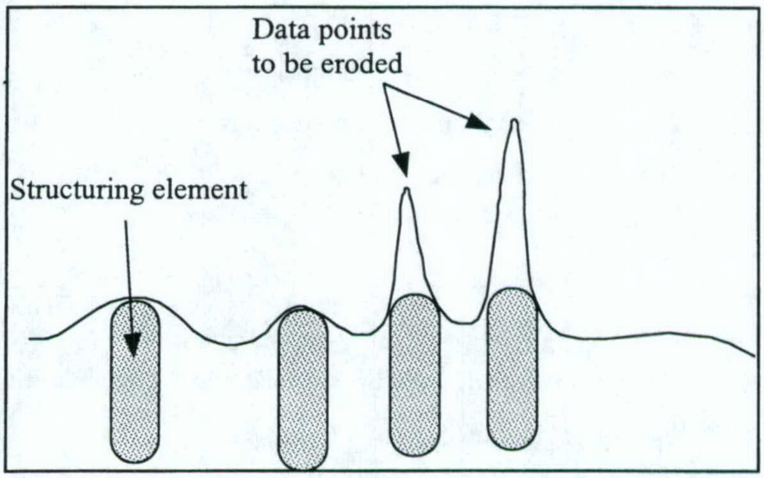


Figure 16: Erosion of surface data

exit This command ends INTEGRATE. Pressing F12, clicking on "exit" in the function key menu, or selecting the exit command from the right mouse key menu also ends INTEGRATE.

Usage: exit

Example: exit

Result: The INTEGRATE session ends when the user clicks the left mouse button in the "Exit or Continue" window.

eye This command changes the perspective from which the user views the objects on the screen. Eye does not change any object's position in the axis system; rather, the viewer's "eye" moves to the specified position in the axis system.

Eye needs three parameters, the X, Y, and Z location of the eye. Eye 0 700 0 puts the viewer's eye on top of the object. Eye 0 0 700 puts the viewer's eye in front of the object. Eye 700 0 0 puts the viewer's eye beside the object.

Usage: eye (distance along X in mm) [distance along Y in mm] [distance along Z in mm]

Example: eye 0 0 300

Result: The "eyepoint" (the location of the viewer in the screen axis system) is moved to a point 300 mm from (and facing) the origin, along the Z axis.

eyedist

This command resets the default eye distance to the specified distance. The default distance is set at 700. When it is changed, the **front, back, left, side, right, bottom**, and **top** commands use the new distance as the default distance for computing eye position.

Usage: eyedist (distance)

Example: eyedist 300

Result: The eye distance is set to 300 mm greater than the default eye distance of 700 mm. When a perspective command (front, right, top, etc.) is executed, the image on the screen appears as if the viewer's "eye" is 1000 mm from the object.

fcmod

This command separately removes all red, green, and blue color components which fall outside of the specified boundaries. This can be useful in identifying the color components of various features of the object, facilitating the development of color-based detection or segmentation methods.

Usage: fcmod (U/L)(R/G/B)(value) [...]

Example: fcmod LR32 UR128

Result: All red components below 32 or above 128 are set to 0, removing the red component from that surface point.

fcwrite

This command writes out an ASCII fullcolor (24-bit) file. It requires a single argument which is the base name of the file to be written. The suffix ".color" is appended to the base name.

Usage: fcwrite filename

Example: fcwrite head

Result: An ASCII file called head.color is created which contains the color components for each vertex.

fill [on/off]

This command enables or disables automatic void fill for an object after a command, such as cload or resample, which might create new voids in the data.

Usage: toggle command

Example: fill

Result: INTEGRATE's automatic fill function is enabled or disabled.

filter

This command filters the data with one of the INTEGRATE smoothing filters. Select a type of smoothing filter and a scale factor to determine the strength of the filter. (The larger the scale, the larger the number of adjacent points involved in the filter function.) Options are: GAUSSIAN, DISCRETE, or GREEN filters. The filter may be applied latitudinally, longitudinally or in both directions. Note: When using **filter** on a trimmed area, points outside the area are used in calculations. This may result in shrinkage from the rest of the data. See **filtseg**.

Usage: filter (GAUSS/DISCRETE/GREEN) (scale) (LAT/LON/BOTH)

Example: filter gauss 3 lat

Result: INTEGRATE smoothes the active object latitudinally.

filtseg This command is identical to the filter command except that the edge of a trimmed area is replicated and used in place of data outside the area. This helps to prevent shrinkage.

Usage: filtseg {GAUSS/DISCRETE/GREEN} (scale) {LAT/LON/BOTH}

Example: filtseg gauss 3 lat

Result: INTEGRATE smoothes the active object latitudinally with minimum distortion at the edges of the smoothed area.

fix_seam This command corrects any mismatch between the sides of the seam where the end of the data set meets the beginning. The mismatch is caused by subject movement during the scan. Fix_seam operates on a grid-type active object only. Note: For best results, make sure the object's trim boundaries correspond with the physical seam.

Usage: fix_seam

Example: fix_seam

Result: INTEGRATE corrects seam mismatch in the active object.

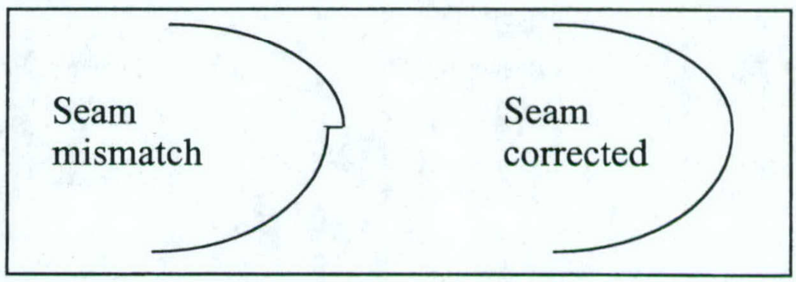


Figure 17: Seam correction with fix_seam

fk This command allows dynamic re-definition of the functions in the function key menu at the top of the screen. It requires 3 parameters: the modifier/key ID, whether the key is a command (c) or a partial command (p), and the text string of the command to be executed or placed in the echo buffer. The Modifier/key ID is either F#, S#, C#, or A#, where F designates an unmodified function key, S designates a function key with the Shift key, C designates a function key with the Ctrl key, and A designates a function key with the Alt key, and # represents a function key number from 1 to 12.

Usage: fk (F/S/C/A)# (c/p) (command text)
Example: fk a6 c auto_jaws
Result: The command at column F6, row A in the function key menu is changed to "auto_jaws". The next time Alt-F6 is pressed or the left mouse button is clicked over "auto_jaws", INTEGRATE will try to execute the auto_jaws command.

fkeys This command turns the function key menu display on or off. It works the same as the other toggle commands.

Usage: toggle command
Example: fkeys
Result: The function key menu display at the top of the screen appears or disappears.

force_lnd [on/off]

Normally, INTEGRATE only displays landmarks within the trim boundaries of an object. This command overrides the trim boundaries to always display all landmarks.

Usage: toggle option

Example: force_lnd

Result: The display of landmarks for this object will change from a trim-limited display to an unlimited display.

front

This command moves the viewer's "eye" to the front of the object. Front has one optional parameter: a distance. If the distance is positive, the viewer's eye will be positioned that much further than the currently set eye distance from the object. If the distance is negative, the viewer's eye will be positioned that much closer to the object.

Usage: front [distance]

Example: front 300

Result: The viewer is now looking at the front of the object from a distance of 300 mm beyond the currently set eye distance (see eyedist).

fullcolor

This command allows the use of all available color information (up to eight bits each of red, green, and blue) instead of the abbreviated color (four bits of red and green, three bits of blue) normally available. Fullcolor requires one parameter: the base file name of the color file. INTEGRATE automatically adds a .color or .rgb extension to the filename you specify. The data path is also added to the filename. Fullcolor has two optional parameters: min and max. If min and max are specified, the color information is modified as follows: for each color axis (r, g, b) all values below min are set to 0, all values above max are set to 255, and all values between min and max are rescaled to the range 0 to 255. For example, FULLCOLOR 001_53P 0 128 tells INTEGRATE to read all color information from file 001_53P.RGB (or 001_53P.COLOR) and rescale all values from 0 to 128 to the range 0 to 255 (values of 128 become 255, values of 64 become 128, etc.).

Usage: fullcolor base_color_file_name [min max]
Example: fullcolor 101_53p
Result: INTEGRATE reads all color information from file 101_53p.

gcv

This command makes a logarithmic plot of the Generalized Cross Validation equation for head scan data. Generalized Cross Validation is a method for determining the best discrete Gaussian filter scale for the given data. The minimum value of the plot is a conservative estimate for the best scale. INTEGRATE also calculates other statistics (mean, standard deviation) to help determine the appropriate scale.

Usage: gcv [defaults] or [start steps_decade total_steps]
Example: gcv
Result: INTEGRATE makes a logarithmic plot of the GCV equation for the active object.

gload

This command reads in a MOVIE.BYU geometry file. Gload takes one argument: the name of the MOVIE.BYU geometry file to be loaded.

Usage: gload movie_file [land_file]
Example: gload body_scan.g
Result: INTEGRATE loads body_scan.g.

gouraud [on/off]

This command toggles gouraud shading on or off while in RGB mode (see the rgb command). Gouraud shading presents a smoother-looking image.

Usage: toggle command
Example: gouraud
Result: The active object appears with gouraud shading in RGB mode.

gwrite

This command writes out a MOVIE.BYU geometry file. Gwrite takes one argument, the name of the MOVIE.BYU file to be written out.

Usage: gwrite movie_file

Example: gwrite body_scan.g

Result: INTEGRATE writes body_scan.g to the directory from which INTEGRATE was launched.

help [on/off]

This command turns the list of available commands on or off.

Usage: toggle command

Example: help

Result: The command list appears or disappears.

hide

This command temporarily removes an object from the screen without removing it from the object pool. Hide has one optional parameter: the number of the object to hide. If an object number is not specified, INTEGRATE hides the active object.

Usage: hide [object number]

Example: hide 5

Result: Object 5 disappears, but remains in the object pool.

histogram	This command creates a histogram of an object. The histogram is limited to ten equally spaced, user specified intervals. INTEGRATE stores the histogram in a file called histogram.dat. INTEGRATE then activates a full-screen editor. The editor displays the histogram so it can be edited. Usage: histogram obj interval Example: histogram 1 30 Result: An editor window appears containing the histogram of object 1 in intervals of 30 mm.
ident	This command adds a tag line to the measurement file to allow later identification of the exact value being measured. It takes one parameter: the tag line to put into the file. Usage: ident <identification string> Example: ident Sitting Height Result: A line is added to the measurement file identifying the next reading as the Sitting Height.
interpolate	This command fills "voids" (holes/missing data) in a grid dataset. It has three options: linear, smooth, or gauss <scale>. Linear interpolation is equivalent to the "do fill" command, smooth produces the most accurate results, and gauss (gaussian) produces the nicest looking results. Interpolate is extremely slow and should be used only when necessary. Usage: interpolate {linear|smooth|gauss (scale)} Example: interpolate smooth 5 Result: holes in the active (grid) object are filled using the smooth fill algorithm.

intersect

This command tests whether two objects have intersecting surfaces in their current positions. Intersect works with the active object and takes one parameter: the object number of the object to test for intersection with the active object. Because intersect tests every polygon on one object against every polygon on the other object, it is extremely slow for very large objects, but performs acceptably when one or both objects are small.

Usage: intersect (obj#)
Example : intersect 4
Result : The active object (in its current position) is compared against object 4 (in its current position) and INTEGRATE reports whether there is any intersection between the objects.

intrplnd

This command enables or disables interpolated landmark mode. Normally, landmark picking selects the point most representing the average of the points in the pick region, but with intrplnd mode enabled, the actual average (when pickmode average is selected) is set as the landmark point.

Usage: toggle command
Example: intrplnd
Result: Interpolated landmark mode is enabled.
Landmarks will be positioned at the average coordinates of the points within the pick window (cursor boundaries) if PICKMODE AVERAGE is active.

jaw

This command only works when the active object is a calipers object (calipers on). It moves either jaw 1 or jaw 2 forward or backward along the slide bar. Positive movement is along the direction from jaw 1 to jaw 2. Negative movement is along the direction from jaw 2 to jaw 1.

Usage: jaw [1|2] <distance>

Example: jaw 2 -10

Result: If jaw 2 is more than 10 mm from jaw 1, it will be moved 10 mm closer to jaw 1 along the slide bar. If it is less than 10 mm from jaw 1, it will be moved to adjoin jaw 1.

jump

This command acts as a "go to" command in a script file. Jump can be dependent on a condition. Conditions currently supported are: always (always jump), count (jump a specified number of times), and smooth (jump based on iterative smoothing criteria).

Jump needs two parameters: the condition for the jump, and the comment line in the script file to go to.

Usage: jump condition [comment identifier]

Example: jump count 5 * start here

Result: When INTEGRATE reaches the jump command in the script file, INTEGRATE goes to the line containing "* start here" and begins executing commands at that point. INTEGRATE executes the jump five times.

land [on/off]	This command displays or hides the landmark points for the active object (if landmarks have been read in for this object). Usage: toggle command Example: land Result: The active object's landmarks appear or disappear.
landlist [on/off]	This command turns the list of standard landmarks on or off. If the active object has assigned landmarks, the screen/world coordinates of the standard and auxiliary landmarks for the active object will also be displayed when landlist is enabled. Usage: toggle command Example: landlist Result: The landmark list appears or disappears.
left	This command moves the viewer's "eye" to the left side of the active object. Usage: left [distance] Example: left Result: The viewer is now looking at the left side of the object. Left obeys the same distance settings as the other eyepoint commands.
lload	This command loads a landmark file for the active object. Lload needs one parameter: the name of the landmark file to be loaded. Usage: lload (landmark file name) Example: lload 010_53p.lnd Result: INTEGRATE loads the 010_53p.lnd landmark file.

lmlist This command toggles between the first half and the second half of the auxiliary landmark list when "landlist" is selected. This allows information on twice as many auxiliary landmarks to be viewed, but if the list of auxiliary landmarks is more than twice as long as screen space allows, there will still be landmarks whose information is not visible.

Usage: toggle option
Example: lmlist
Result: When the landmark list is displayed, the list will toggle between the first half and the second half of the list.

lregister This command registers (aligns) an object to another object by least-squares fitting of corresponding standard landmarks.

Lregister needs two parameters: the number of the object to be registered and the reference object number.

Usage: lregister (obj) (ref_obj)
Example: lregister 3 2
Result: INTEGRATE registers object 3 to object 2, effectively translating and rotating object 3 into the position of object 2.

lwrite

This command writes a new landmark file. Lwrite has one required parameter and one optional parameter. The name of the landmark file to be written is required. Rotate (or just r), the optional parameter, rotates and translates the XYZ coordinates of the object from the native object axis system into screen/world coordinates before writing the file.

Usage: lwrite file_name [rotate]

Example: lwrite new_landmarks.lnd r

Result: INTEGRATE writes the landmark file new_landmarks.lnd to the directory from which INTEGRATE was launched, rotating and translating the object's XYZ coordinates into screen/world coordinates.

man

This command displays the file "users_guide.txt" on the screen. To turn the manual off, follow the directions for exiting the editor activated by this command. For this command to work properly, users_guide.txt must exist either in the directory where INTEGRATE was started, or in a directory pointed to by the environment variable INTEGRATE.

On UNIX/Linux systems, "setenv INTEGRATE /home/integrate" will allow INTEGRATE to find a copy of users_guide.txt in /home/integrate, regardless of the directory from which INTEGRATE was started.

Usage: man

Example: man

Result: A reference list of all INTEGRATE commands appears in a window on the screen.

median

This command replaces a point with the median value of the points in its neighborhood. This helps eliminate data spikes.. Median requires one parameter: the size of the neighborhood window for median computation.

Usage: median window_size

Example: median 20

Result: For every point in the active object, INTEGRATE computes a median value and assigns that value to the point. INTEGRATE uses the 20 surrounding points for the calculation.

merge

This command will merge the points from two objects to create a third object. Merge requires 2 parameters: the numbers of the objects to be merged. An optional third parameter specifies whether to use the maximum radius, (max), the minimum radius (min), or the average radius (avg) in areas where the objects overlap. Merge resamples both objects before merging them, so it is slow and it produces a grid object.

Usage: merge (object1) (object2) [min,max,avg]

Example: merge 1 2 avg

Result: The radial values from object 1 and object 2 will be averaged and saved in the next available object number.

mirror

This command reverses the coordinates of the active object, creating a mirror image of the object. Presently only the data points are mirrored; landmarks and contours are not mirrored. For grid objects, the mirror plane is always the XY plane. For mesh objects, the mirror plane is the plane perpendicular to an optionally specified axis. If no axis is specified, the YZ plane is used.

Usage: mirror [axis]
Example: mirror
Result: The active object is mirrored about either the XY plane (grid object) or the YZ plane (mesh object).

mload

This command loads a saved displacement matrix. Mload requires one parameter: the name of the saved file.

Usage: mload (matrix file)
Example: mload head_scan.mtx
Result: INTEGRATE loads the head_scan.mtx displacement matrix and transforms the active object into a new coordinate system.

modland

This command positions the landmark pointer on a specified element in the landmark list. When picking points, modland allows the user to return to any landmark in the list and reassign coordinates for that landmark. Modland's parameter indicates which landmark is to be picked or re-picked next.

Usage: modland (standard land number)
Example: modland 32
Result: The landmark pick list pointer moves directly to standard landmark 32. L32 will be the next landmark picked.

move

This command moves the active object along the X, Y, and/or Z axes. Move needs three parameters: the distance in millimeters to move the object along each axis. These distances will be added to the object's current position. The current position appears in the blue box in the lower left corner of the screen.

Usage: move (distance along X) [distance along Y] [distance along Z]
Example: move 0 100
Result: The active object moves 100 mm up the Y axis.

move_vertex

This command allows the movement of individual mesh object vertices. It requires up to four parameters: the vertex number to move (see pickmode point), and the X, Y, and Z distances to move the vertex. This is useful for hand-editing specific objects when the object is inaccurate and no automated method is available for correcting it.

Usage: move_vertex (vertex number) (X distance) [Y distance] [Z distance]
Example: move_vertex 11234 1 2 3
Result: Vertex 11234 moves 1 mm along the X axis, 2 mm along the Y axis, and 3 mm along the Z axis.

movie_segment

This command extracts a segment of a mesh object by specifying bounding planes in the directions of the X, Y, and Z axes. A complete bounding box can be specified by l(x/y/z)#, which provides the lower boundary on the specified axis and u(x/y/z)#, which provides the upper boundary on the specified axis. All six possible boundaries can be specified, with positive or negative values, as appropriate. Note: The actual screen position of the object (not its native coordinate system) determines which points will be copied. "movie_seg -" can be used to copy a complete object.

Usage: movie_segment (l(x/y/z)#/u(x/y/z)#)) ...
Example: movie_segment lx-100 ux100 ly-100 uy100 lz-100 uz100
Result: INTEGRATE copies all the points within a 200 mm cube around the origin of the active object and stores the copied points in a new object.

mwrite

This command writes the most recent displacement matrix of the active object to a specified file.

Usage: mwrite file
Example: mwrite head_scan.mtx
Result: INTEGRATE writes the active object's displacement matrix to a file called head_scan.mtx. INTEGRATE stores the file in the directory from which INTEGRATE was started.

nameland

This command assigns a name to an auxiliary (Z) landmark. The form of the command is: nameland (Z#) (new_landmark_name). New_landmark_name must not contain blanks.

Usage: nameland (Zlandmark#) (new_name)
Example: nameland z2 helmet_landmark1
Result: Auxiliary landmark Z2 is renamed helmet_landmark1.

negsub

This command performs the point-by-point subtraction of each corresponding radial value on two objects. Specify a reference object and a replace object of the same size (that is, with the same trim values), and INTEGRATE subtracts one object from the other. The second object is subtracted from the first object, and the results less than zero (negative radial values) are retained. INTEGRATE stores the subtraction results in the replace (second) object.
Note: For best results, **register** and **resample** the objects before executing negsub.

Usage: negsub (reference_obj) (replace_obj)
Example: negsub 2 1
Result: INTEGRATE subtracts object 1 from object 2 and stores all negative values in object 1.

newcenter

The command changes the center point of an object to the specified coordinates or landmark.

Usage: newcenter { X Y Z | L# | Z#}
Example: newcenter z2
Result: The new center of rotation of the active object will be at auxiliary landmark z2.

newland This command allows manual entry of a landmark. It operates on the active object and requires 4 parameters: the destination landmark number (L# or Z#) and the X (right-left), Y (up-down), and Z (near-far) coordinates in the screen coordinate system. The user can include a name for the landmark, also.

Usage: newland (Z#/L#) (X) [Y] [Z] [new landmark name]
Example: newland z10 20 20 20 helmet2
Result: INTEGRATE creates a new auxiliary landmark, Z10, at the specified coordinates and names it helmet2.

new_meas This command renames or reinitializes the measurement storage file. It has one parameter: the name of the new measurement file. If the parameter is omitted, the current measurement storage file is deleted.

Usage: new_meas (new file name)
Example: new_meas DataFile.txt
Result: Any data in the file DataFile.txt is deleted and the next measurement to be stored will be stored in DataFile.txt

new_order This command reads in a file with a new order for landmark picking for the 42 standard landmarks or a subset of those landmarks. One parameter is required: the file name of the file with the new order typed in it. The file should have landmark numbers (separated by a space) in the order that the landmarks are to be picked. This command requires an active object to run.

Usage: neworder filename
Example: neworder special_landmarks
Result: INTEGRATE reads in the landmark order file called special_landmarks.

new_vertex This command permits the addition of a vertex to a triangular mesh file. It requires up to three parameters: the screen/world x, y, and z coordinates of the new vertex. The world coordinates are converted to object coordinates internally.

Usage: new_vertex (x) (y) (z)

Example : new_vertex 100 200 300

Result: A new vertex is added to the active object at world coordinates (100, 200, 300).

open This command activates a dialog box for loading a file. It provides the standard directory and file lists to allow the user graphically to select the file to be opened. It currently supports the following file types: ply files (gzipped or not) (pload), matrix files (mload), landmark files (lload), and Cyberware Echo files (cload).

Usage: open

Example: open

Result: The file open/load dialog box is activated.

option

This command controls preset options which determine the initial state of an object after it is loaded, or in some cases after it has been transformed. Option may be followed by any object-level on/off command. For example, OPTION WIREFRAME OFF will cause all future objects to be loaded without immediately displaying the wireframe form. Similar options could be OPTION SURFACE ON to turn on the surface form of an object as soon as it is loaded, or OPTION FILL ON to cause voids to be automatically filled as part of the object load process, and after any significant manipulation of an object.

Usage: option (object toggle command) (on/off)
Example: option land off
Result: When you load an object and its landmark file, the object appears with the landmarks hidden.

ortho

This command toggles between the normal perspective (objects shrink with increasing distance) view and an orthographic (objects do not shrink with distance) view of the object space. This is useful for removing parallax from the view to better interpret relationships between points.

Usage: toggle command
Example: ortho
Result: Objects toggle from normal (perspective) view to orthographic view or from orthographic view to normal (perspective) view.

pause

This command supports script processing by stopping the execution of a script file until the operator presses a key to continue. Note: a script file can also be paused while it is running by pressing any key. **Do not use Escape (Esc) to pause a script file. Escape terminates script processing.**

Usage: pause

Example: pause

Result: When INTEGRATE reaches the pause command in the script file, command execution stops until the operator presses a key. Note: **Do not press Escape (Esc) to continue. Escape terminates script processing.**

pick [on/off]

This command enables or disables point picking mode. When pick mode is enabled, all parts of objects farther away than the origin of the world space are hidden to make picking easier and to reduce the chance of picking points on the far side of an object by accident.

Usage: toggle command

Example: pick

Result: Point picking mode is enabled or disabled.

pickmode

This command sets the point picking mode to perform specific operations. It has two parameters: a mode for selecting a specific point near the cursor, and a mode for using the point to automatically perform an operation.

Selection options are: centroid, closest, or median.

Centroid chooses the average longitude and latitude of all the points which were detected in the pick region (the points indicated by the cursor).

Closest chooses the longitude and latitude of the point in the pick region closest to your eye position.

Median chooses the longitude of the longitudinal median point and latitude of the latitudinal median point in the pick region.

Usage: pickmode (centroid/closest/median)
Example: pickmode closest
Result: When the left mouse button is clicked, INTEGRATE chooses the longitude and latitude of the point in the pick region closest to the viewer's "eye."

pickmode (continued)

Operation options are: con2p, con3p, cir2p, cir3p, mul2p, mul2a, land, auxland, distance, gdist, delpnt, delall, addpoly, delpoly, and point.

Con2p causes every odd point to be the start point of a contour, and every even point to be the end point of a contour, with the center of the object defining the plane of the contour.

Con3p causes every group of three points to define a contour plane, with the contour running from the first point to the second point.

Cir2p and *cir3p* work the same as con2p and con3p, except that they create a complete circumference. (Note that contours and circumferences do not work properly if they encounter a boundary of the object.)

Mul2p creates multiple, consecutive two-point contours (the third point is the object center), with the second point of a contour becoming the first point of the next contour.

Mul2a creates multiple, consecutive two-point contours (the third point is the Y-axis at mid-latitude), with the second point of a contour becoming the first point of the next contour.

Land creates a new standard landmark from every selected point.

pickmode (continued)

Auxland creates a new auxiliary landmark from every selected point.

Distance computes the straight-line distance from the first selected point to each point selected thereafter.

Gdist computes the straight-line distance from a selected point on one object to a selected point on a different object. Each point must be selected with the object of interest being the active object.

Delpnt deletes the selected point.

Delall deletes all points within the pick window.

Addpoly adds a new polygon to the active object after the user clicks on three vertexes.

Delpoly removes a polygon from the active object after the user clicks on the three vertexes forming the polygon.

Point reports the longitude, latitude, radius, and XYZ value for each selected point.

Usage: pickmode (con2p/con3p/cir2p/cir3p/mul2p/mul2a/land/auxland/distance/gdist/delpnt/delall/addpoly/delpoly/point)
Example: pickmode auxland
Result: INTEGRATE creates an auxiliary landmark at the next point where the user picks by clicking the left mouse button.

planes [on/off] This command turns the XY, YZ, and XZ reference planes on or off.

Usage: toggle command

Example: planes

Result: The reference planes appear or disappear.

pload This command loads a polygon mesh file in Stanford .ply format. After loading, the object is identical to a movie.byu-format mesh. All commands that work with movie.byu files can be applied to the loaded object. Pload requires one parameter: the name of the .ply file to be loaded.

Usage: pload (file name)

Example: pload 052.ply

Result: INTEGRATE loads the polygon mesh file called 052.ply.

points [on/off] This command enables or disables a display of the scan data for the active object as individual points.

Usage: toggle command

Example: points

Result: INTEGRATE turns the active object's point display on or off.

pop This command redraws the **INTEGRATE** window over any other windows. It is equivalent to the window menu POP option.

Usage: pop

Example: pop

Result: Other open windows on the screen disappear behind the INTEGRATE window.

possub This command performs a subtraction on two grid objects along each radial value. Specify a reference object and a replace object of the same size (that is, with the same trim values), and INTEGRATE subtracts one object from the other. The second object is subtracted from the first object, and the results greater than zero (positive radial values) are retained. INTEGRATE stores the subtraction results in the replace (second) object.

Note: For best results, **register** and **resample** the objects before executing possub.

Usage: possub (reference_obj) (replace_obj)
Example: possub 2 1
Result: INTEGRATE subtracts object 1 from object 2 and stores all positive radial values in object 1.

print This command executes a tool that saves a copy of the INTEGRATE window for editing, display, and/or printing. It requires one parameter: the base name of the file to use in saving the window image. Due to a lack of a universal screen capture utility in the WIN32 environment, this command does not work under WIN32. If such a utility is available, then it should be possible to capture the INTEGRATE window through other methods.

Usage: print (file base name)
Example: print
Result: A file called (file base name).xwd is created with an image of the INTEGRATE window. A utility such as GIMP can be used to prepare the image for display and/or printing.

pshrink This command shrinks an object by a specified percentage.

Usage: pshrink (percentage) [shrinkage center coordinates]
Example : pshrink 25
Result : The active object is reduced to ¾ of its original size (100%-25%). Since no center coordinates were specified, it is shrunk around the center of mass of the object.

push This command causes all other windows on the screen to appear on top of the **INTEGRATE** window. It is equivalent to the window menu PUSH option.

Usage: push
Example: push
Result: Open windows on the screen appear on top of the INTEGRATE window.

pwrite This command writes out a polygon mesh object in Stanford .ply format. Pwrite requires one parameter: a name for the file to be written.

Usage: pwrite (file name)
Example: pwrite 101.ply
Result: INTEGRATE writes the polygon mesh object to a file called 101.ply and stores the file in the directory from which INTEGRATE was started.

readout This command only works with a calipers object which is the active object. It produces the measured distance between the 2 jaws of the calipers. A parameter indicates whether the measurement is for the outside of the object (readout outside) (for example, measuring head breadth), or the inside of the object (readout inside) (for example, measuring the distance between a hand and the body).

Usage: readout [inside|outside]
Example: readout outside
Result: The distance between the inner surface of the jaws is reported in a screen message.

recolor

This command rescales the indexed color file values for the active object to maximize the available color information. Use recolor to make an object lighter or darker when it's displayed in indexed color (RGB mode off). Recolor requires two parameters: the minimum color to distinguish from black and the maximum color to distinguish from white. All colors between the min and the max will be rescaled to evenly fill the color space between black and white. Note that a negative minimum is equivalent to adding a positive offset to all color values.

The minimum color is usually set to zero. If the user specifies a maximum value less than 256, the object appears lighter; if the user specifies a maximum color greater than 256, the object appears darker.

(lighter $\Leftarrow$ 256 $\Rightarrow$ darker)

Usage: recolor min max

Example: recolor 0 198

Result: The active object becomes lighter when displayed in indexed color.

refresh

This command supports script processing by redrawing the screen in the middle of a sequence of scripted operations. Normally the screen is not redrawn during a scripted sequence.

Usage: refresh

Example: refresh

Result: When INTEGRATE reaches the refresh command in a script file, INTEGRATE redraws the objects on the screen.

remark

This command inserts a text string in the session audit trail. The text string appears in the INTEGRATE session record, stored in the directory from which INTEGRATE was started.

Usage: remark (string)

Example: remark starting new session

Result: INTEGRATE inserts "starting new session" in the session record.

resample

This command creates a new grid object from the active object with a new set of cylindrical (grid) coordinates.

Resample accepts up to four optional parameters that affect mesh objects only during resampling: a longitude angle resolution (in radians), a latitude resolution (in mm), a maximum radius (in mm) for scaling radial values, and the number of interpolated points to include in the sampling process. Grid objects retain the same resolutions when resampled.

The longitude and latitude resolution parameters determine how many points will be produced in the resampling.

The maximum radius parameter determines the resolution of the radial values at each cylindrical point. The larger the radius the lower the resolution. If too small a radius is specified, unexpected results may occur.

If the number of extra samples (oversampling factor) is not specified, it is set to 4. The available range is from 0 to 16. Oversampling generally produces better results than using the original points only.

Usage: resample (lon res) {lat res) {max radius) (samples)
Example: resample 0.01 0.5 130 16
Result: INTEGRATE creates a copy of the active object with .01 radian resolution (628 samples per row), .5 mm vertical resolution, a maximum radius of at least 130 mm, and 16x oversampling, transforming it to the screen/world coordinate system.

rgb

This command toggles between color map mode (limited to 2048 colors) and RGB mode (full 24-bit color). When the fullcolor command is applied to an object, the full color is available for viewing the surface when in RGB mode.

Usage: toggle command
Example: rgb
Result: The active object appears in full color.

right

This command moves the viewer's "eye" to the right side of the object. Right has one optional parameter: a distance. If the distance is positive, the viewer's eye will be that much further away from the object than the default distance. If the distance is negative, the viewer's eye will be that much closer to the object.

Usage: right [distance]
Example: right
Result: The viewer sees the right side of the object.

rotate

This command rotates the active object around the X, Y, and/or Z axes. Rotate needs three parameters: the angle to rotate the active object around each of the three axes. These angles will be added to the current position. The current position is shown in the blue box in the lower left corner of the screen.

Usage: rotate (degrees around X) (degrees around Y) (degrees around Z)
Example: rotate 0 30 20
Result: INTEGRATE rotates the active object 30 degrees around the Y axis (counterclockwise) and 20 degrees around the Z axis (counterclockwise).

ruin

This command randomly creates void patches in an object. The command requires both the object and a copy of the object to operate. After execution, the copy of the object will contain only the data of the newly created voids. It is used for testing various object editing tools.

Usage: ruin (object to ruin) (copy object)
Example: ruin 3 4
Result: INTEGRATE creates voids in object 3 and stores the voided data in object 4.

save

This command activates a "File Save" dialog box to allow the user to graphically specify the location of a file to be saved.

Usage: save
Example: save
Result : A File Save dialog box appears.

scwrite

This command writes out the standard (indexed 11-bit) color for the active object to a file. It requires one parameter: the name of the file to be written. Note that only grid objects have standard color, and the resulting color file can only be loaded into a grid object with either the recolor (file name must be (geometry file name)) or the fullcolor command (any file name is usable).

Usage: scwrite (file name)
Example: scwrite junk
Results: A file named junk.color is created in SGI Image format.

select

This command selects which object is the active object. Select needs one parameter: the number of the object to be selected. Objects can also be selected by typing in just the object number.

Usage: select (object number)
Example: select 3
Result: Object 3 is now the active object.

set

This command sets a parameter in the ASPEC for an object. Set requires two parameters: a parameter name and a new parameter value. Useful parameter names are RSHIFT, NAME, STUDY, SCAN_TYPE, VERSION, LTOFF, LGOFF, FILLED, AND SMOOTHED. Other names should be used with **EXTREME CAUTION**. The parameter value for NAME, STUDY, or SCAN_TYPE should be a string with no embedded blanks. The value for all other parameter names should be an integer, generally less than 512.

Usage: set (parameter1) (parameter2)
Example: set study traditional
Result: The information contained in the header under STUDY_TYPE will be changed to read "traditional".

shade

This command restores or updates a pseudo-lighting shaded surface to an object.

Usage: shade [object #]
Example: shade
Result: INTEGRATE updates the shading on the active object.

show This command displays an object that has been hidden. Show has one optional parameter: the number of the object to show. If an object number is not specified, INTEGRATE shows the active object.

Usage: show [object #]
Example: show 3
Result: INTEGRATE displays object 3.

show_hid_lnd This command allows landmarks to be visible even through surfaces that would normally obscure them. When show_hid_lnd is enabled, all landmarks within the visibility clipping planes will be displayed. When it is disabled, only those landmark labels that are closer than the nearest object surface will be fully visible.

Usage: toggle command
Example: show_hid_lnd on
Result: all landmark labels on the active object will be visible even if there is a surface between the label and the viewer's eyepoint.

shrink

This command is used to reduce the radial values of a head scan in the mesh (movie.byu or .ply) format uniformly and spherically. An example use of **shrink** is to approximate the inner surface of a hollow object such as a helmet. Shrink has one required parameter: the radial amount to shrink the surface in mm, and three optional parameters: the x, y, and z coordinates of a user-specified center of shrinkage. If no center is specified, the center of mass of the object is used for the shrinkage center.

Usage: shrink (amount in mm) [shrinkage center coordinates]
Example: shrink 10
Result: The head scan will shrink by 10 mm uniformly and spherically around the center of mass of the object.

side

This command moves the viewer's "eye" to the left side of the object. Side has one optional parameter: a distance. If the distance is positive, the viewer's "eye" will be that much further away from the object than the default distance. If the distance is negative, the viewer's "eye" will be that much closer to the object. This command is identical to the left command.

Usage: side [distance]
Example: side
Result: The viewer sees the left side of the object.

skip

This command skips over a landmark slot when picking landmarks. Skip has one optional parameter: the number of landmark slots to skip. If a skip number is not specified, INTEGRATE skips one slot. If a negative number is specified, INTEGRATE skips backward in the landmark list.

Usage: skip [value]
Example: skip -1
Result: INTEGRATE skips backward one slot in the landmark list.

skipto

This command specifies which landmark is to be picked next by resetting the landmark pick number to the specified value. Because pick order can be varied according to a pickorder list with standard landmarks, skipto is most useful for auxiliary landmarks because it directly specifies the number of the next landmark to be picked.

Usage: skipto (landmark number)
Example: skipto 41
Result: If the pickmode is auxland and the active object is using CAESAR landmarks, the pick pointer is positioned at the Left Acromion (Z41).

sleep

This command supports script processing by forcing the script to stop for a given number of seconds, in order to give the operator time to observe the state of an image before processing continues. Sleep accepts one parameter: the number of seconds to wait before continuing. If the number of seconds is not specified, the script pauses for one second. This command is usually preceded by a refresh command.

Usage: sleep [value]
Example: sleep 10
Result: INTEGRATE pauses for 10 seconds when it reaches the sleep command in the script file.

smooth [on/off]

This command enables/disables automatic smoothing for an object after a command, such as cload or resample, which disturbs the smoothness of the data. To execute smoothing for the active object, use **do smooth**.

Usage: toggle command
Example: smooth
Result: Automatic smoothing is turned on or off.

split

This command computes a new landmark at the mid-point between two other landmarks. It operates on the active object and requires three parameters: the destination landmark number (L# or Z#) and the two defining landmarks (L# or Z#). There are two optional parameters for this command, weight1 and weight2, which cause the new landmark to be positioned proportionally between the two defining landmarks. The weight function might be used for determining a combined Center of Gravity from the CG's of two objects of different weights, such as a human head and a helmet system.

Usage: split ([Z/L]destination landmark #) ([Z/L] first landmark #) ([Z/L] second landmark #) [wt1] [wt2]

Example: split z20 z1 z2 .5 2

Result: INTEGRATE creates a new auxiliary landmark, z20, between landmarks z1 and z2. The weights, .5 for z1 and 2 for z2, tell INTEGRATE to position z20 80% of the way toward z2.

• z1 • z20 • z2

INTEGRATE uses the following equation to determine the new landmark's location:

$((weight_1 \bullet coordinates_1)+(weight_2 \bullet coordinates_2))/(weight_1 + weight_2)$

store [on/off]

This command enables or disables storage of data from various measurement commands to a disk file. When store is enabled, results of VOLUME, SURFACE_AREA, DISTANCE, and PICKMODE DISTANCE point picks are stored to "measures.txt" (or another file named by the new_meas command) with appropriate labels.

Usage: toggle command
Example: store on
Result: Data from measurement commands are stored in a file called measures.txt (unless changed by the **new_meas** command) in the directory from which INTEGRATE was launched.

subject [on/off]

This command turns the display for the active object on or off. **Subject on** is the same as **show** and **subject off** is the same as **hide**. **Subject** by itself works like any other on/off command; it toggles between on and off.

Usage: toggle command
Example: subject
Result: INTEGRATE hides or shows the active object.

super

This command allows several objects to be grouped into a “super object” so all the objects can be moved or changed together. In keeping with the concept of a "super-object," this is a "super-command" with six command modifiers: MAKE, ADD, RELEASE, DELETE, LINK, and UNLINK. MAKE creates a new super-object, ADD adds objects to a super-object, RELEASE removes one or more objects from a super-object, DELETE deletes the super-object, LINK attaches a sub-object and applies an offset that causes the sub-object to rotate around the same rotation point as the super-object, and UNLINK removes the offset and detaches the sub-object. Note that the DELETE command (by itself) applied to a super-object is identical to the SUPER DELETE command.

Usage:
SUPER MAKE (sub-obj1) ... (sub-objN)
SUPER ADD (super-obj) (sub-obj1) ... (sub-objN)
SUPER RELEASE (super-obj) (sub-obj1) ... (sub-objN)
SUPER DELETE (super-obj)
SUPER LINK (super-obj) (sub-obj)
SUPER UNLINK (super-obj) (sub-obj)
Example: super make 2 3 4
Result: INTEGRATE groups objects 2, 3, and 4 under a new super object. When the new object is moved, rotated, or hidden, all 3 objects will also be moved, rotated, or hidden.

surface [on/off]

This command enables or disables a display of the scan data for the active object as shaded surface polygons. Depending on the rgb mode and the color data available, the object may be displayed with acquired color data or with false color data such as simulated lighting (shade command) or surface difference information (delta command).

Usage: toggle command

Example: surface

Result: INTEGRATE turns the surface display on or off for the active object.

surface_area

This command computes the displayed surface area of the active object.

Usage: surface_area

Example: surface_area

Result: INTEGRATE computes the surface area of the active object and displays the result.

surf_register

This command attempts to improve a rough registration between 2 similar objects by iteratively computing distances and angles bewteen the 2 surfaces, then correcting for the observed errors. Often a rough alignment can be accomplished using 3 or more common landmarks and using **lregister** or **zregister** to register one object with another.

This command accepts six parameters:

test_object - object to be adjusted based on surface matching

ref_object - oject to be matched against

thres - maximum distance between matched points to limit accidental matching with points on the far side of an object. Should be less than half the minimum diameter of the smallest object, but more than the maximum distance between the two surfaces in the matching region. The default for thresh is 50.0 mm.

nTries - the number of trials between asking the operator for input. After nTries trials, the operator is given the choice of: stopping the registration (N), continuing the registration (Y), or adjusting the number of probes being used (1, 2, 4, 8, 0). The number of probes is the number specified by nProbes, divided by a speedup factor of 1, 2, 4, 8, or 16. The speedup factor makes each loop correspondingly faster, but may reduce the accuracy of the match, so the operator is given control to balance the time spent vs. the accuracy. The default for nTries is 50.

surf_register (continued)

nProbes - the maximum number of probes in each trial. The actual number of probes may be less than nProbes if a speedup factor has been specified. The probes are randomly selected from the entire set of points in the test object. For each probe point, a corresponding point on the reference object is determined by finding the point where the normal to the probe point intersects the reference object. The matched points are then used in the same alignment algorithm used by lregister and zregister. The default for nProbes is 1/10 of the points in the test object.

rotateAxis - sometimes it is desirable to limit the freedom of the rotations and translations which can take place in the matching process. For instance, objects may be known to be positioned at the same height with the same vertical axis, so limiting the displacements to horizontal translations and rotations might be expected to produce a more-exact result. By specifying the vertical axis (usually Y in INTEGRATE) as the rotateAxis, extraneous displacements can be avoided. The default is unconstrained rotations and translations.

Usage: surf_reg (test_obj) (ref_obj) [thres [nTries [nProbes [rotateAxis]]]]

Example: surf_reg 5 4 25.0 15 500

Result: Object 5 will be surface_registered with object 4. Points in object 4 that are more than 25 mm away from a test point in object 1 will be disqualified for surface point matching. Every 15 loops the command will pause for operator action. If object 5 has more than 500 points, the command will choose 500 randomly-selected points for surface matching. RotateAxis is not specified, so rotation and translation are not constrained.

tape

This command converts a surface contour/circumference into a tape-measure-equivalent by eliminating any concave curves in the surface. The resulting distance of the modified contour should then approximate the distance measured with a tape measure.

This command has two parameters: the contour to be modified and has an optional parameter which indicates whether the tape measure completely wraps around the object. This command might not work well for very complex curves, since it uses the center of mass of the contour points as a reference point. If the center of mass falls outside the curve, results are unpredictable.

Usage: tape (contour#) [wrap/nowrap]
Example: tape 2
Result: The contour is converted into a completely convex curve. Since the wrap parameter is not specified, the end points are excluded from the removal algorithm.

text

This command will print a text string on the screen at the specified coordinates. This is useful for documenting screen captures which may be similar and hard to differentiate after capture.

The parameters are x coordinate, y coordinate, size, and the text string. Due to constraints in the supporting graphics library for INTEGRATE 2.x, the size parameter is ignored.

Usage: text (xcoord) (ycoord) (size) (string)
Example: text 10 300 4 my text
Result: The text string "my text" will show up on the lower left hand corner of the screen.

thin

This command sets the frequency of longitude and latitude lines to be shown on the active object. Thinning an object speeds up some INTEGRATE functions.

Thin needs two parameters: the longitude thin factor and the latitude thin factor. For example, for a thin factor of 2, INTEGRATE displays every second data point; for a thin factor of 3, INTEGRATE displays every third data point, and so on.

Usage: thin (lon thin value) (lat thin value)
Example: thin 2 2
Result: INTEGRATE displays only every second data point along each longitude and latitude line.

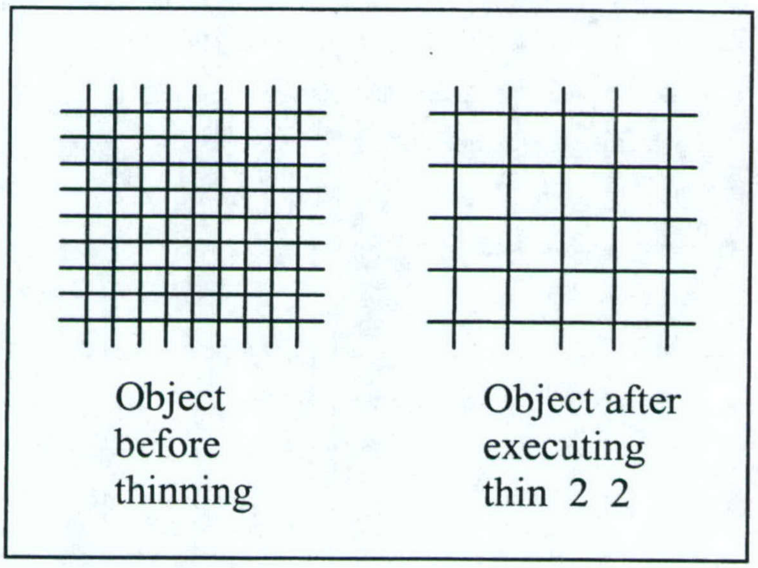


Figure 18: Thinning an object

threshold This command performs a radial threshold operation on a cylindrical grid object. The object and radius threshold values are specified by the user. The qualifiers eq (equal), ne (not equal), lt (less than), le (less than or equal), gt (greater than), ge (greater than or equal) refer to the radii to be zeroed. For example, "threshold 2 lt 55" means all radii in grid object 2 below the 55 threshold should be set to zero.

Usage: threshold (object #) (eq/ne/lt/le/gt/ge) (value)
Example: threshold 4 lt 2
Result: INTEGRATE eliminates all points with radial values less than 2 mm on object 4.

top This command moves the viewer's "eye" to the top of the object.

Top has one optional parameter: a distance. If the distance is positive, the viewer's eye will be positioned that much further from the object. If the distance is negative, the viewer's eye will be positioned that much closer to the object.

Usage: top [distance]
Example: top 200
Result: The viewer now sees the top of the active object, 200 mm further away from the object than before.

toupee

This command fills in the top of the head of the active object. Note that this command works best when the object is positioned so that the (estimated!) highest point on the head is centered on the Y axis. This command needs two parameters: the lowest latitude for the toupee, and the highest latitude for the toupee. Check the object coordinates in the blue box on the lower left for the coordinates to use for the toupee. Note that the specified latitudes must be within the current trim area.

Usage: toupee (bottom of toupee latitude) (top of toupee latitude)
Example: toupee 196 203
Result: INTEGRATE places a cap or "toupee" on the void on top of the active object.

transparent [on/off]

This command makes the surface display for an object partially transparent, allowing visualization of the detail of the grid or of inner objects.

Usage: toggle command
Example: transparent
Result: The active object becomes semi-transparent.

trim This command modifies the starting and ending longitudes and latitudes of a grid object so that only the necessary part of the active object will be displayed or operated on by other commands. Trim needs four parameters which will change the starting longitude, ending longitude, starting latitude, and ending latitude. These parameters will be added to the current values. To reduce a longitude and latitude boundary, use a negative number. The current starting and ending longitude and latitude are displayed in the blue box in the lower left corner of the screen.

Usage: trim (left_long) (right_long) (lower_lat) (upper_lat)
Example: trim 30 -100 50 -50
Result: INTEGRATE trims the active object.

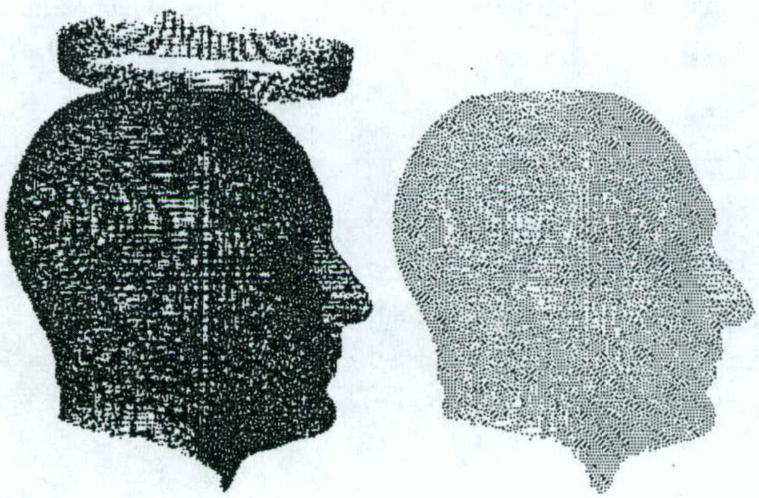


Figure 19: Trimming noise from the top of an object: before trimming (the object on the left) and after trimming (the object on the right)

volume

This command computes the volume inside the surface of the active object. This command also generates the coordinates of the center of volume. If the object has numerous voids or has a surface where any horizontal line drawn from the Y-axis can intersect more than one point on the surface, this command will produce an incorrect volume result.

Usage: volume

Example: volume

Result: INTEGRATE computes the volume and center of volume of the active object and displays the result in the lower left corner of the screen.

wait [on/off]

Many of INTEGRATE's commands stop and wait for user acknowledgement of a notification of completion or an error message. This command allows the user to disable the wait for acknowledgement.

Usage: toggle command

Example: wait off

Result: INTEGRATE will not wait for acknowledgement after reporting a result or an error.

walls This command sets the clipping planes. The clipping planes control the size of the viewing area.

Walls needs two parameters: near or far.

near - points closer to the eye than the near value will not be displayed (initial value 100).

far - points farther from the eye than far will not be displayed (initial value 1400).

Walls also accepts two other parameters: full and half. "Walls full" automatically sets the near wall to 100 and the far wall to twice the distance of the viewpoint from the origin of the grid. "Walls half" automatically sets the near wall to 100 and the far wall to the distance of the viewpoint from the origin of the grid, eliminating from view the back half of an object centered on the origin. If the viewpoint is moved using eye, top, front, or side, the walls may need to be adjusted to prevent clipping of the object.

Usage: walls (near distance) (far distance)

Example: walls 698 702

Result: INTEGRATE limits the viewing area to the space between 698 and 702 mm from the eyepoint.

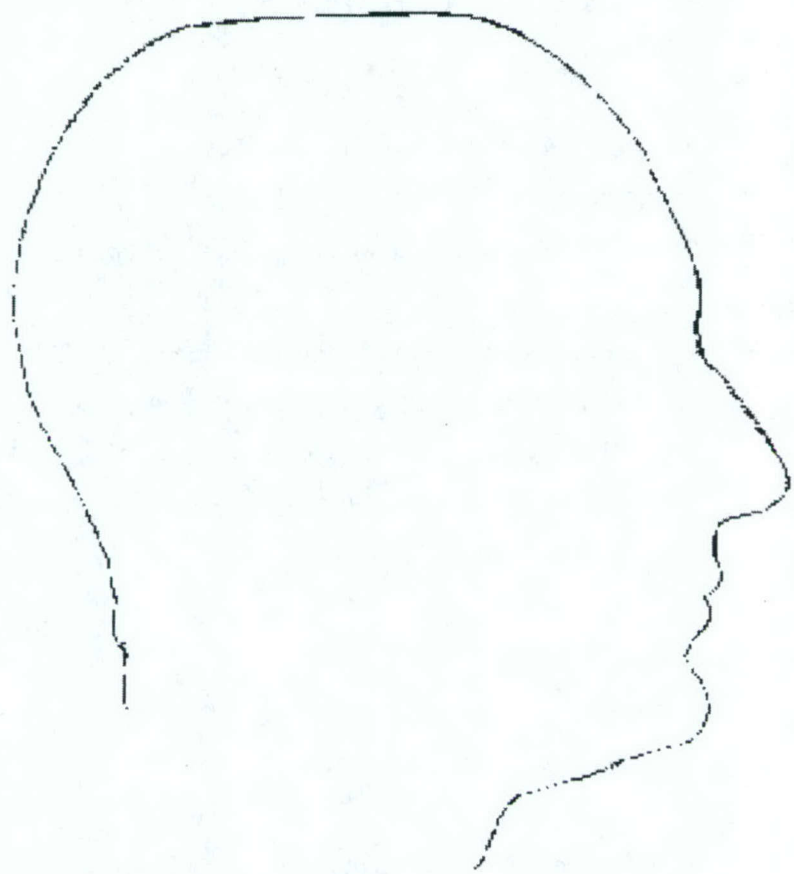


Figure 20: With walls set to 698 702, only a cross section of the object appears

white This command sets the screen background color to white. Landmark and object points will change colors so that they will show up against the white background. Typically white is used to prepare a screen for printing to reduce the total number of pixels which must be transferred to the printer.

Usage: white

Example: white

Result: The screen background turns white.

wireframe [on/off] This command enables or disables wireframe display of the active object.

Usage: toggle command
Example: wireframe
Result: Wireframe display of the active object turns on or off.

wload This command reads in a polygon mesh file in wavefront .obj format. After loading, the object is identical to a movie.byu-format mesh. All commands that work with movie.byu files can be applied to the loaded object. Wload requires one parameter: the name of the .obj file to be loaded.

Usage: wload (file name)
Example: wload sample.obj
Result: The wavefront-format object in file sample.obj is loaded in.

wwrite This command writes out a polygon mesh file in wavefront .obj format. Wwrite requires one parameter: the name of the file to be written.

Usage: wwrite (file name)
Example: wwrite 025.obj
Result: INTEGRATE writes the mesh object to a wavefront-format file called 025.obj and stores the file in the directory from which INTEGRATE was started.

xload This command reads in data stored in XYZ coordinate format (see xwrite).

Usage: xload file

Example: xload head_scan.xyz

Result: INTEGRATE reads in the file head_scan.xyz.

xwrite This command writes scan data XYZ coordinates to an ASCII (text) file. If the -g (grid) option is specified, longitude, latitude, and radius are also written to the file. If the -a (all) option is specified, void points are written to the file; otherwise, only non-void (radius > 0.0) points are written. If the -w (waterline) option is specified, the data points are written in latitude-major order (all points at the same latitude grouped together) instead of longitude-major order. If no options are specified, the filename to be written is the first parameter. If options are specified, the filename to be written is the second parameter.

Usage: xwrite [-agw] file

Example: xwrite -a ascii52.2.xyz1

Result: INTEGRATE writes data, including void points, to a file called ascii52.2.xyz1, stored in the directory from which INTEGRATE was started.

zload

This command reads in previously stored contour data (see zwrite). Zload requires one parameter: the name of the contour file to be read in. Note: if a contour file exceeds the maximum allowed size of a single contour line, the contour will be broken into separate contour lines as required.

Usage: zload (file name)
Example: zload contours
Result: INTEGRATE reads in the contour data file called contours.

zregister

This command registers an object to another object by least-squares fitting of corresponding auxiliary landmarks. Zregister needs two parameters: the number of the object to be registered and the reference object number.

Usage: zregister (obj) (refobj)
Example: zregister 2 1
Result: INTEGRATE uses the auxiliary landmarks to align object 2 to object 1.

zwrite

This command writes out the points which make up one or more contours or circumference lines. The first parameter to zwrite is the name of the file to be written. All following parameters are contour numbers to be written. The optional rotate parameter causes INTEGRATE to convert the contour coordinates from the object axis system into the screen/world axis system before writing.

Usage: zwrite (file name) (contour1) ... (contourN) [rotate]
Example: zwrite contour_file 3 4 5 r
Result: INTEGRATE writes contours 3, 4, and 5 to a file called contour_file in the directory from which INTEGRATE was started. The contour coordinates are converted from native object coordinates into screen/world coordinates before writing.

5.0: INTEGRATE'S AUDIT TRAIL FUNCTION

INTEGRATE's Audit Trail maintains a record of all user commands entered during an INTEGRATE session. The Audit Trail file allows the user to:

- analyze an INTEGRATE session to discover the cause of unsatisfactory results,
- record the history of a modified dataset so future users can evaluate the validity of the final data,
- create a script file that will automatically reproduce the results of the session.

INTEGRATE stores the commands in a file called AUDITFILE.xxxx, where xxxx can be up to 30 characters at the beginning of the name of the INTEGRATE system host. For example, a system whose host name is falcon will produce audit trail files called AUDITFILE.falcon. The audit trail files are stored in the directory from which INTEGRATE was launched. In the event that WIN32 does not produce a host name, the name "Win32Host" will be used.

INTEGRATE stores all commands executed during the INTEGRATE session except actions initiated during point picking. For example, if the user deleted points from a dataset with **pickmode delall**, the deletion of every point in the dataset would not appear in the audit file. If all the pickmode actions were included, it would be difficult to find more useful information in the file.

IAUDIT is a program that allows the user to view and manipulate INTEGRATE's audit trail files. Instructions for using IAUDIT appear below.

5.1: Using IAUDIT

Follow these steps to view and manipulate audit trail files:

1. Change directory (cd) to the directory from which INTEGRATE was started for the relevant session.
2. Type **iaudit** and press Enter.
3. When the list of audit trail files appears, determine which audit file contains the relevant session record. The sessions are numbered and are listed by date, time, and user name. A few lines of the session list might look like this:

    42: *** Integrate Session 3/15/1996 8:55:32 chris

43: *** Integrate Session 3/18/1996 10:15:12 mark

44: *** Integrate Session 3/19/1996 14:39:10 josephine

4. Type an IAUDIT command that contains the option for carrying out the required action. The table below defines the IAUDIT options:

iaudit (with no options)	Lists the INTEGRATE sessions by date and time.
-a	Lists every command that was executed in the session or sessions.
-b	Creates a script file that includes all "first level" commands. Commands from subsidiary script files are not included, though the @(script file name) commands that launch scripts are included.
-d	Deletes sessions from the session list.
-t	Creates a script file with any subordinate scripts merged in to create a single-level script containing all commands executed during the session.

5. To **list all the commands** in sessions 5 through 10, type a command that looks like this:

   iaudit -a 5 10

6. To **create a script file** from the commands in sessions 24, type a command that looks like this:

   iaudit -b 24>newScript

   where "newScript" is the name of the script file to be created.

7. To **delete sessions** 10 through 20 from the session list, type a command that looks like this:

   iaudit -d 10 20

8. To **create a script file** from the commands in session 12, **merging any secondary script file commands**, type a command that looks like this:

   iaudit -t 12>newScript

   where "newScript is the name of the script file to be created.

9. Use a text editor to edit the scripts. Some commands may need to be deleted, and some commands may need to be combined into one command.

6.0: REFERENCES

1. Brunsman, M.A., Files P.S. The CG Dataset: Whole Body Surface Scans of 53 Subjects, AL/CF-TR-1996-0160, Armstrong Laboratory, Wright-Patterson Air Force Base, Oh.

2. Brunsman, M.A., Daanen, H., and Files P.S. Earthquake in Anthropometry: The View from the Epicenter. *CSERIAC Gateway*, Volume VII: No.2, 1996.

3. Daanen, H., Brunsman, M.A., Taylor, S.E., and Nurre, J.H. Absolute Accuracy of the Cyberware WB4 Whole Body Scanner, AL/CF-TR-1997-0046, Armstrong Laboratory, Wright-Patterson Air Force Base, Oh.

4. McConville, J.T., Clauser, C.E., Churchill, T. D. , Cuzzi, J and Kaleps, I. (1980). Anthropometric Relationships of Body and Body Segment Moments of Inertia, Air Force Aerospace Medical Research Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, OH.

5. Nurre, J.H., Whitestone, J.J., Hoffmeister, J.W., and Burnsides, D.B. (1995), Removing Impulse Noise From Human Head Scan Data, AL/CF-TR-1995-0054, Armstrong Laboratory, Wright-Patterson Air Force Base, Oh.

6. Robinette, K.M., and Whitestone, J.J. (1994) The Need for Improved Anthropometric Methods for the Development of Helmet Systems. Aviation, Space, and Environmental Medicine, pp. A95-A99, May 1994, Aerospace Medical Association, Alexandria, VA.

7. Robinette, K.M. and Whitestone, J.J. (1992). Methods for Characterizing the Human Head for the Design of Helmets, AL-TR-1992-0061, Crew Systems Directorate, Human Engineering Division, Armstrong Laboratory, Wright-Patterson Air Force Base OH.

8. Whitestone JJ, Slemker TC, Ause-Ellias KL, Richard RL, Miller, S. (1995), Fabrication of Total Contact Burn Masks -- Employing Human Body Topography and Computer-Aided Design and Manufacturing (CAD/CAM), Journal of Burn Care and Rehabilitation, Burn Science Publishers, Inc.

9. Whitestone, J.J. (1993), Design and Evaluation of Helmet Systems Using 3D Data. Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting - 1993, Vol. 1, pg. 63, The Human Factors and Ergonomics Society, Santa Monica CA.

10. Whitestone, J.J. and Robinette, K.M. (1992). High Resolution Human Body Surface Data for the Design of Protection Equipment. Proceedings of the 2nd Pan Pacific Conference on Occupational Ergonomics, Safety and Environmental Protection Research Institute, MMI, Wuhan, China.

APPENDIX A: TUTORIALS: IMAGE DATA AND SCRIPT FILES

FILES NEEDED FOR TUTORIAL 1

SCRIPT FILE	IMAGE FILE(S)	COLOR FILE(S)	LANDMARK FILE(S)
tutorial_1.txt	010_53p	010_53p.rgb	010_53p.lnd

FILES NEEDED FOR TUTORIAL 2

SCRIPT FILE	IMAGE FILE(S)	COLOR FILE(S)	LANDMARK FILE(S)
tutorial_2.txt	010_53p	010_53p.rgb	010_53p.lnd
	010_53ph	010_53ph.rgb	010_53ph.lnd
	53psize5	53psize5.rgb	53psize5.lnd

FILES NEEDED FOR TUTORIAL 3

SCRIPT FILE	IMAGE FILE(S)	COLOR FILE(S)	LANDMARK FILE(S)
tutorial_3.txt	010_53p	010_53p.rgb	010_53p.lnd

FILES NEEDED FOR TUTORIAL 4

SCRIPT FILE	IMAGE FILE(S)	COLOR FILE(S)	LANDMARK FILE(S)
tutorial_4.tzt	53psize5	53psize5.rgb	53psize5.lnd
	100_53ph	100_53ph.rgb	100_53ph.lnd
	100_53p	100_53p.rgb	100_53p.lnd
	101_53ph	101_53ph,rgb	101_53ph.lnd
	101_53p	101_53p.rgb	101_53p.lnd
	102_53ph	102_53ph.rgb	102_53ph.lnd
	102_53p	102_53p.rgb	102_53p.lnd
	104_53ph	104_53ph.rgb	104_53ph.lnd
	104_53p	104_53p.rgb	104_53p.lnd
	105_53ph	105_53ph.rgb	105_53ph.lnd
	105_53p	105_53p.rgb	105_53p.lnd

FILES NEEDED FOR TUTORIAL 5

SCRIPT FILE	IMAGE FILE(S)
tutorial_5.txt	comp1a1.thorax.cdd
	tno1a1.thorax.cdd

FILES NEEDED FOR TUTORIAL 6

SCRIPT FILE	IMAGE FILE(S)	LANDMARK FILE(S)
tutorial_6.txt	head.g pasgt.g head.pasgt.cdd	head.lnd pasgt.lnd head.pasgt.lnd

FILES NEEDED FOR TUTORIAL 7

SCRIPT FILE (S)	IMAGE FILE(S)	COLOR FILE(S)	MATRIX FILE(S)
tutorial_7.txt	tsa_stda.ply		std2.mtx, waist_circ.mtx
	c50x25.g	calipers.g.color	caliper.mtx

FILES NEEDED FOR TUTORIAL 8

SCRIPT FILE (S)	IMAGE FILE(S)	LANDMARK FILE(S)
tutorial_8.txt	csr0099a.ply[.gz]	csr0099a.lnd

FILES NEEDED FOR TUTORIAL 9

SCRIPT FILE (S)	IMAGE FILE(S)	LANDMARK FILE(S)
tutorial_9.txt	csr0099a.ply[.gz]	csr0099a.lnd

FILES NEEDED FOR TUTORIAL 10

SCRIPT FILE (S)	IMAGE FILE(S)	LANDMARK FILE(S)
tutorial_10.txt loadLink.txt	csr0099a.head.ply csr0099a.neck.ply csr0099a.bodyLimbs.ply	csr0099a.joints.lnd

APPENDIX B: HEAD AND FACE ANATOMICAL LANDMARKS: DESCRIPTIONS AND ILLUSTRATIONS

ANATOMICAL LANDMARK DEFINITIONS

CHELION: the corners of the mouth formed by the juncture of the lips.

ECTOCANTHUS: the outer corners of the eyes; the lateral canthus

ENDOCANTHUS: the inner corners of the eyes; the medial canthus

FRONTOTEMPORALE: The point of deepest indentation of the temporal crest from the frontal bone above the browridges.

GLABELLA: Landmark title for the most forward point in the midline of the forehead between the brow ridges.

GONION: A corner of the jaw; the lateral point of the corner of the mandible (jaw bone).

INFRAMALAR: The most inferior point of the zygomatic process of the maxilla.

INFRAORBITALE: The lowest point on the inferior margin of the orbit or eye socket.

INFRAZYGION: The inferior border of the zygomatic arch directly below zygion.

MENTON (LANDMARK): Title for the inferior point of the mandible (tip of the chin) in the midsagittal plane.

NUCHALE: The lowest bony point on the base of the back of the skull in the mid-sagittal plane.

PROMENTON: The most anterior projection of the soft tissue of the chin in the midsagittal plane.

PRONASALE (LANDMARK): Title for the tip of the nose.

PUPIL: The center of the contractile (usually round) aperture in the iris of the eye; the center of the pupil.

SELLION: The point of greatest indentation of the nasal root depression. (the point of greatest indentation where the bridge of the nose meets the forehead.)

STOMION: The point of contact between the upper and lower lips in the midsagittal plane.

SUBMANDIBULAR: Under the mandible or lower jaw.

SUBNASALE: The point inferior to the nose where the base of the nasal septum meets the philtrum; the point of the intersection of the groove of the upper lip (philtrum) with the inferior surface of the nose in the midsagittal plane.

SUPRAECTOCANTHUS: The most protruding point of the browridge located on the same vertical axis as ectocanthus.

SUPRAENDOCANTHUS: The most protruding point of the browridge located on the same vertical axis as ectocanthus.

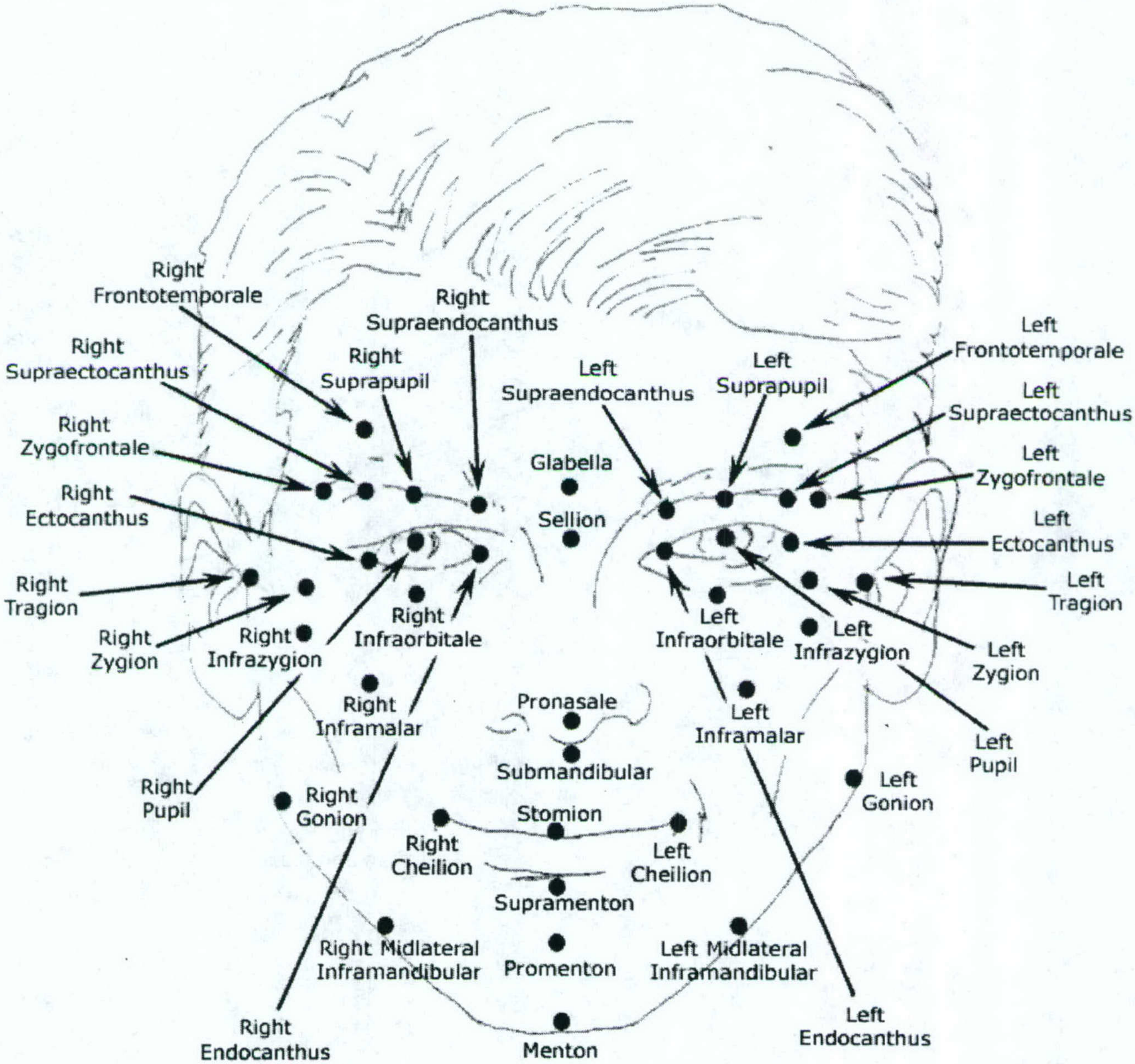
SUPRAMENTON: The point of greatest indentation of the mandibular symphysis.

SUPRAPUPIL: The most protruding point of the browridge located on the same vertical axis as the corresponding right or left pupil.

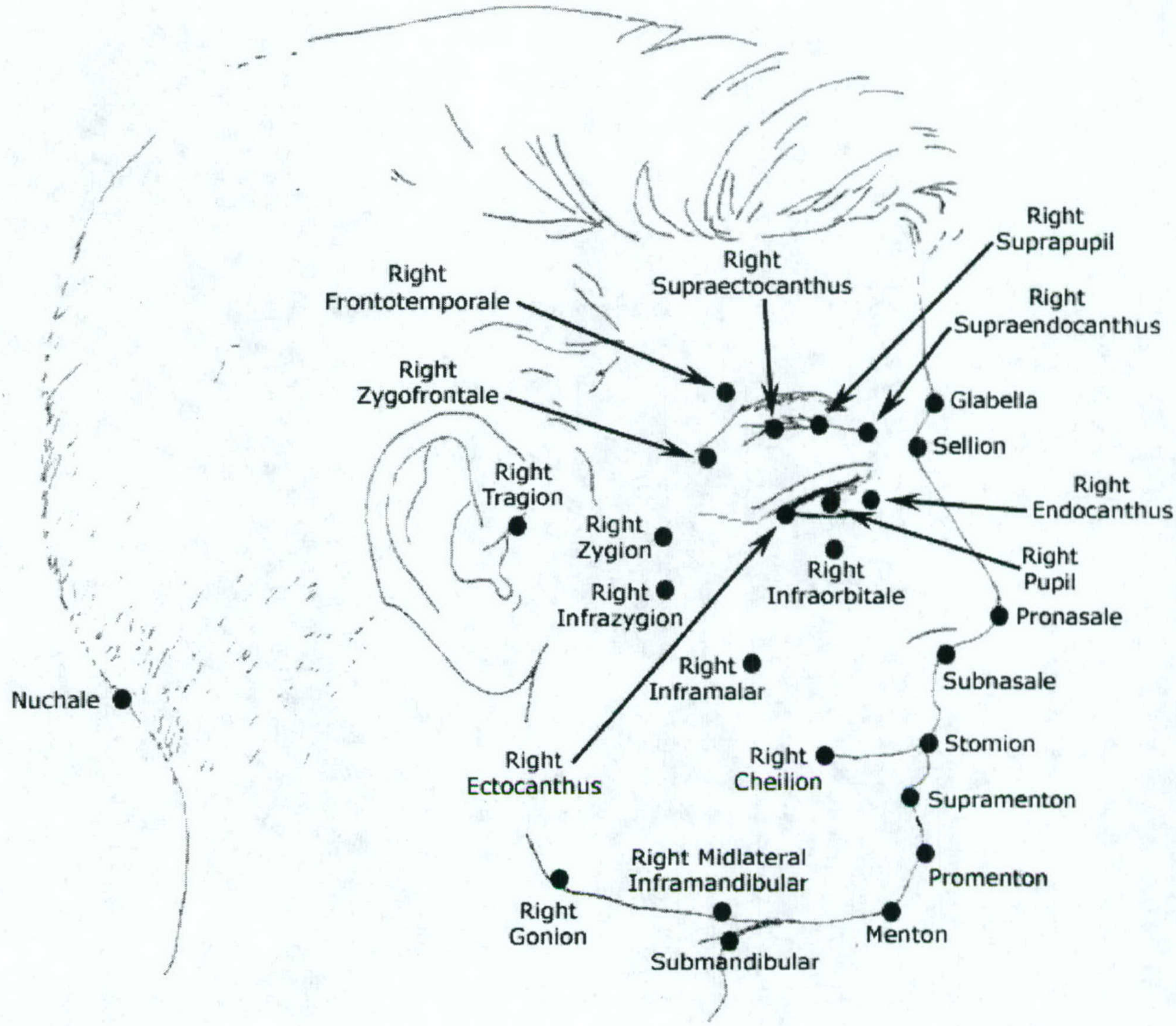
TRAGION: point located at the notch just above the tragus of each ear. this point corresponds approximately to the upper edge of the ear hole (external auditory meatus) of the skull.

ZYGION: the lateral point of the zygomatic arch.

ZYGOFRONTALE: the most lateral point of the frontal bone where it forms the upper margin of the bony eye socket.



Not Visible: Nuchale, Submandibular


Right
Suprapupil
Right
Frontotemporale
Right
Supraectocanthus
Right
Supraendocanthus
Right
Zygofrontale
Glabella
Sellion
Right
Tragion
Right
Endocanthus
Right
Zygion
Right
Pupil
Right
Infraorbitale
Right
Infrazygion
Pronasale
Right
Inframalar
Subnasale
Nuchale
Stomion
Right
Ectocanthus
Right
Cheilion
Supramenton
Right Midlateral
Inframandibular
Promenton
Right
Gonion
Menton
Submandibular

APPENDIX C: WHOLE BODY ANATOMICAL LANDMARKS: DESCRIPTIONS AND ILLUSTRATIONS

3-D Landmark List for Standing Posture as used in the CAESAR Survey

#	Segment	View	Landmarks			Method
1)	Head	Front	Sellion			scanned
2)			Infraorbitale	(R)		scanned
3)			Infraorbitale	(L)		scanned
4)			Supramenton			scanned
5)			Tragion	(R)		scanned
6)			Gonion	(R)		scanned
7)			Tragion	(L)		scanned
8)			Gonion	(L)		scanned
9)		Back	Nuchale			scanned
10)	Torso	Front	Clavicale	(R)		scanned
11)			Suprasternale			scanned
12)			Clavicale	(L)		scanned
13)			Thelion/Bustpoint	(R)		scanned
14)			Thelion/Bustpoint	(L)		scanned
15)			Substernale			scanned
16)			Tenth Rib	(R)		scanned
17)			ASIS	(R)		scanned
18)			Tenth Rib	(L)		scanned
19)			ASIS	(L)		scanned
20)		F/B	Iliocristale	(R)		scanned
21)			Trochanterion	(R)		scanned
22)			Iliocristale	(L)		scanned
23)			Trochanterion	(L)		scanned
24)		Back	Cervicale		(Spine I)	scanned
25)			Tenth Rib Midspine		(Spine II)	scanned
26)			PSIS	(R)		scanned
27)			PSIS	(L)		scanned
28)			Preferred Waist, Posterior			scanned
29)		F/B	Acromion	(R)		scanned
30)		Front	Anterior Axilla Ref. Point		(R)	scanned
31)	Wrist		Radial Styloid	(R)		scanned
32)	Torso	Back	Posterior Axilla Ref. Point		(R)	scanned
33)	Arm		Olecranon	(R)		scanned
34)			Lateral Humeral Epicondyle		(R)	scanned
35)			Medial Humeral Epicondyle		(R)	scanned
36)			Radiale	(R)		scanned
37)	Hand	Front	Metacarpal-Phalangeal II		(R)	scanned
38)	Hand	Side	Dactylion	(R)		scanned
39)	Wrist	Back	Ulnar Styloid	(R)		scanned

40)	Hand		Metacarpal-Phalangeal V	(R)	scanned
41)	Torso	F/B	Acromion (L)		scanned
42)		Front	Anterior Axilla Ref. Point	(L)	scanned
43)	Wrist		Radial Styloid (L)		scanned
44)	Torso	Back	Posterior Axilla Ref. Point	(L)	scanned
45)	Arm		Olecranon (L)		scanned
46)			Lateral Humeral Epicondyle	(L)	scanned
47)			Medial Humeral Epicondyle	(L)	scanned
48)			Radiale (L)		scanned
49)	Hand	Front	Metacarpal-Phalangeal II	(L)	scanned
50)		F/B	Dactylion (L)		scanned
51)	Wrist	Back	Ulnar Styloid	(L)	scanned
52)	Hand		Metacarpal-Phalangeal V	(L)	scanned
53)	Knee		Knee Crease (R)		scanned
54)		F/B	Lateral Femoral Epicondyle	(R)	scanned
55)			Medial Femoral Epicondyle	(R)	scanned
56)	Foot		Metatarsal-Phalangeal V	(R)	scanned
57)	Ankle		Lateral Malleolus	(R)	scanned
58)			Medial Malleolus	(R)	scanned
59)			Sphyrion (R)		scanned
60)	Foot		Metatarsal-Phalangeal I	(R)	scanned
61)		Back	Posterior Calcaneous	(R)	scanned
41)		Front	Phalange II (Foot)	(R)	scanned
63)	Knee		Knee Crease (L)		scanned
64)		F/B	Lateral Femoral Epicondyle	(L)	scanned
65)			Medial Femoral Epicondyle	(L)	scanned
66)	Foot		Metatarsal-Phalangeal V	(L)	scanned
67)	Ankle		Lateral Malleolus	(L)	scanned
68)			Medial Malleolus	(L)	scanned
69)			Sphyrion	(L)	scanned
70)	Foot		Metatarsal-Phalangeal I	(L)	scanned
71)		Back	Posterior Calcaneous	(L)	scanned
72)		Front	Phalange II (Foot)	(L)	scanned
73)	Torso	Front	Crotch		calculated


Z41
Z29
Z32
Z22
Z23
Z21
Z67
Z66
Z56

APPENDIX D: LANDMARK FILES: ANATOMICAL AND AUXILIARY LANDMARKS FOR THE HEAD AND FACE

```
SUBJECT_ID = 010_53p
SCAN_TYPE  = NO_TYPE
STUDY_NAME = * * *  NO STUDY  * * *
LAND_STUDY = New Study
STD_LAND = 42
AUX_LAND = 0
STANDARD =
  1 371 116    72.66  -11.54  181.31  -71.74      (Right Tragion)
  2 343 124    74.03  -35.69  193.81  -64.86      (Right Zygion)
  3 344 118    72.86  -34.34  184.43  -64.26      (Right Infra-Zygion)
  4 346  78    55.40  -24.91  121.91  -49.49      (Right Gonion)
  5 283  67    68.70  -64.97  104.72  -22.35      (Right Mid-Infra-Mandibular)
  6 299 153    83.38  -72.03  239.14  -41.99      (Right Frontotemporale)
  7 297 146    86.20  -75.52  228.20  -41.57      (Right Zygofrontale)
  8 280 110    90.75  -86.84  171.93  -26.35      (Right Infra-Malar)
  9 288 137    80.96  -74.80  214.13  -30.99      (Right Ectocanthus)
 10 259 138    75.82  -75.77  215.69   -2.79      (Right Endocanthus)
 11 281 129    84.46  -80.52  201.63  -25.51      (Right Infraorbitale)
 12 237 161    89.27  -86.86  251.64   20.62      (Glabella)
 13 237 143    93.29  -90.76  223.51   21.55      (Sellion)
 14 233 122   117.98 -113.31  190.69   32.85      (Pronasale)
 15 234 111   102.96  -99.23  173.49   27.46      (Subnasale)
 16 234  75   106.66 -102.80  117.22   28.44      (Promenton)
 17 231  65    92.67  -88.35  101.60   27.98      (Menton)
 18 257  93    96.21  -96.20  145.36   -1.18      (Right Chelion)
 19 234  96   106.02 -102.18  150.05   28.27      (Stomion)
 20 216  94    99.29  -87.57  146.92   46.80      (Left Chelion)
 21 234  53    58.81  -56.68   82.84   15.68      (Submandibular)
 22 196 130    84.98  -62.97  203.19   57.07      (Left Infraorbitale)
 23 215 138    75.36  -66.02  215.69   36.34      (Left Endocanthus)
 24 186 137    80.62  -52.66  214.13   61.04      (Left Ectocanthus)
 25 182  70    68.10  -41.90  109.41   53.69      (Mid-Infra-Mandibular)
 26 176 154    84.04  -46.69  240.70   69.88      (Left Frontotemporale)
 27 175 148    85.42  -46.58  231.32   71.60      (Left Zygofrontale)
 28 189 110    91.67  -62.39  171.93   67.16      (Left Infra-Malar)
 29 132 123    77.00   -3.78  192.25   76.91      (Left Zygion)
 30 133 117    76.22   -4.68  182.87   76.08      (Left Infra-Zygion)
 31 128  81    61.23    0.00  126.60   61.23      (Left Gonion)
 32 108 118    74.80   18.17  184.43   72.56      (Left Tragion)
 33 495  93    85.38   83.53  145.36  -17.68      (Nuchale)
 34 271 139    80.98  -79.62  217.26  -14.83      (Right Pupil)
 35 286 148    88.59  -82.65  231.32  -31.89      (Right SupraEctocanthus)
 36 269 153    86.10  -85.01  239.14  -13.68      (Right SupraPupil)
 37 256 150    87.00  -87.00  234.45    0.00      (Right SupraEndocanthus)
 38 203 140    84.67  -67.39  218.82   51.27      (Left Pupil)
 39 192 150    88.79  -62.79  234.45   62.78      (Left SupraEctocanthus)
 40 205 153    89.22  -72.31  239.14   52.26      (Left SupraPupil)
 41 219 150    87.33  -78.48  234.45   38.30      (Left SupraEndocanthus)
 42 234  87   104.15 -100.38  135.98   27.78      (SupraMenton)
AUX =
END =
```

```
SUBJECT_ID = 010_53ph
SCAN_TYPE  = NO_TYPE
STUDY_NAME = * * *  NO STUDY  * * *
LAND_STUDY = New Study
STD_LAND = 42
AUX_LAND = 6
STANDARD =
  1 -999 -999    0.00    0.00    0.00    0.00
  2 -999 -999    0.00    0.00    0.00    0.00
  3 -999 -999    0.00    0.00    0.00    0.00
  4 -999 -999    0.00    0.00    0.00    0.00
  5 -999 -999    0.00    0.00    0.00    0.00
  6 279 130  109.69 -105.35  203.19  -30.55      (Right Frontotemporale)
  7 280 123  109.70 -104.98  192.25  -31.85      (Right Zygofrontale)
  8 271  86  109.52 -107.67  134.42  -20.05      (Right Infra-Malar)
  9 -999 -999    0.00    0.00    0.00    0.00
 10 -999 -999    0.00    0.00    0.00    0.00
 11 268 104  107.21 -106.05  162.55  -15.74      (Right Infraorbitale)
 12 232 130  123.92 -118.59  203.19   35.97      (Glabella)
 13 232 114  121.49 -116.26  178.18   35.26      (Sellion)
 14 231  90  136.23 -129.87  140.67   41.14      (Pronasale)
 15 -999 -999    0.00    0.00    0.00    0.00
 16 231  49  111.87 -106.65   76.59   33.78      (Promenton)
 17 -999 -999    0.00    0.00    0.00    0.00
 18 -999 -999    0.00    0.00    0.00    0.00
 19 -999 -999    0.00    0.00    0.00    0.00
 20 -999 -999    0.00    0.00    0.00    0.00
 21 -999 -999    0.00    0.00    0.00    0.00
 22 201 105  106.89  -83.45  164.12   66.79      (Left Infraorbitale)
 23 -999 -999    0.00    0.00    0.00    0.00
 24 -999 -999    0.00    0.00    0.00    0.00
 25 -999 -999    0.00    0.00    0.00    0.00
 26 185 128  111.11  -71.54  200.06   85.02      (Left Frontotemporale)
 27 185 121  111.76  -71.96  189.12   85.51      (Left Zygofrontale)
 28 192  88  109.96  -77.76  137.54   77.75      (Left Infra-Malar)
 29 -999 -999    0.00    0.00    0.00    0.00
 30 -999 -999    0.00    0.00    0.00    0.00
 31 -999 -999    0.00    0.00    0.00    0.00
 32 -999 -999    0.00    0.00    0.00    0.00
 33 -999 -999    0.00    0.00    0.00    0.00
 34 -999 -999    0.00    0.00    0.00    0.00
 35 -999 -999    0.00    0.00    0.00    0.00
 36 -999 -999    0.00    0.00    0.00    0.00
 37 -999 -999    0.00    0.00    0.00    0.00
 38 -999 -999    0.00    0.00    0.00    0.00
 39 -999 -999    0.00    0.00    0.00    0.00
 40 -999 -999    0.00    0.00    0.00    0.00
 41 -999 -999    0.00    0.00    0.00    0.00
 42 -999 -999    0.00    0.00    0.00    0.00
AUX =
  1 387 122  121.91    4.49  190.69 -121.83 Helmet Land 1
  2 354  94  115.02  -41.39  146.92 -107.32 Helmet Land 2
  3 233 171  123.48 -118.60  267.27   34.39 Helmet Land 3
  4 115  92  114.98   18.26  143.80  113.52 Helmet Land 4
  5  83 118  122.87   64.46  184.43  104.61 Helmet Land 5
  6 493  80  107.78  104.87  125.04  -24.90 Helmet Land 6
END -
```

APPENDIX E: LANDMARK FILES: ANATOMICAL LANDMARKS FOR THE WHOLE BODY

Landmark Files: Anatomical Landmarks

For The Whole Body (as used in the CAESAR Survey)

```
SUBJECT_ID = vstemplate.ply
SCAN_TYPE  = NO TYPE
STUDY_NAME = * * *  NO STUDY  * * *
LAND_STUDY = New Study
STD_LAND = 0
AUX_LAND = 73
STANDARD =
AUX =
 1   0   1   36.96    7.57   36.18  704.00 Sellion
 2   0   2   67.98    4.91   67.80  676.00 Rt. Infraorbitale
 3   0   3   56.79   50.38   26.21  674.00 Lt. Infraorbitale
 4   0   4   47.98   12.05   46.44  614.00 Supramenton
 5   0   5  163.57   23.01  161.94  696.00 Rt. Tragion
 6   0   6  156.42   32.22  153.07  618.00 Rt. Gonion
 7   0   7  134.57  125.96   47.35  698.00 Lt. Tragion
 8   0   8  126.97  115.59   52.54  628.00 Lt. Gonion
 9   0   9  230.56  150.84  174.37  694.00 Nuchale
10   0  10  108.38   31.66  103.65  518.00 Rt. Clavicale
11   0  11   86.94   47.38   72.90  509.00 Suprasternale
12   0  12   92.35   68.53   61.91  526.00 Lt. Clavicale
13   0  13  150.17  -99.57  112.42  354.00 Rt. Thelion/Bustpoint
14   0  14   95.29   50.07  -81.08  360.00 Lt. Thelion/Bustpoint
15   0  15   39.70  -39.03   -7.27  276.00 Substernale
16   0  16  158.23  -79.09  137.05  226.00 Rt. 10th Rib
17   0  17  168.83  -73.40  152.04   72.00 Rt. ASIS
18   0  18  111.31   86.85  -69.62  216.00 Lt. 10th Rib
19   0  19  111.16   87.03  -69.15   76.00 Lt. ASIS
20   0  20  255.79  -20.82  254.94  136.00 Rt. Iliocristale
21   0  21  247.16  -38.04  244.22   22.00 Rt. Trochanterion
22   0  22  205.38  198.07  -54.32  144.00 Lt. Iliocristale
23   0  23  216.12  209.28  -53.95   24.00 Lt. Trochanterion
24   0  24  234.76  151.94  178.96  590.00 Cervicale
25   0  25  218.28  154.31  154.38  210.00 10th Rib Midspine
26   0  26  258.96  144.89  214.64   96.00 Rt. PSIS
27   0  27  247.74  215.93  121.44   96.00 Lt. PSIS
28   0  28  227.91  165.51  156.68  133.00 Waist, Preferred, Post.
29   0  29  313.27    9.59  313.12  522.00 Rt. Acromion
30   0  30  253.53  -51.03  248.34  390.00 Rt. Axilla, Ant
31   0  31  392.35 -205.27  334.37  -40.00 Rt. Radial Styloid
32   0  32  331.73   49.49  328.02  378.00 Rt. Axilla, Post.
33   0  33  394.48  -27.79  393.50  206.00 Rt. Olecranon
34   0  34  394.42  -69.00  388.34  218.00 Rt. Humeral Lateral Epicn
35   0  33  352.36   -3.34  352.34  192.00 Rt. Humeral Medial Epicn
36   0  36  395.09  -78.22  387.27  202.00 Rt. Radiale
37   0  37  429.33 -260.07  341.60 -126.00 Rt. Metacarpal Phal. II
38   0  38  466.50 -262.21  386.37 -240.00 Rt. Dactylion
39   0  39  434.47 -180.53  395.19  -52.00 Rt. Ulnar Styloid
40   0  40  435.27 -183.83  394.55 -128.00 Rt. Metacarpal-Phal. V
41   0  41  249.04  248.23  -20.01  536.00 Lt. Acromion
42   0  42  203.82  190.41  -72.72  402.00 Lt. Axilla, Ant
43   0  43  352.71  222.28 -273.85  -24.00 Lt. Radial Styloid
44   0  44  279.31  279.19    8.03  388.00 Lt. Axilla, Post.
45   0  45  329.12  313.48 -100.24  224.00 Lt. Olecranon
46   0  46  329.80  297.56 -142.21  244.00 Lt. Humeral Lateral Epicn
47   0  45  293.85  284.46  -73.68  209.00 Lt. Humeral Medial Epicn
```

48	0	46	329.30	292.17	-151.91	228.00	Lt. Radiale
49	0	49	379.92	209.99	-316.61	-104.00	Lt. Metacarpal-Phal. II
50	0	50	437.41	255.26	-355.20	-204.00	Lt. Dactylion
51	0	51	374.50	282.49	-245.87	-32.00	Lt. Ulnar Styloid
52	0	52	387.68	283.57	-264.36	-108.00	Lt. Metacarpal-Phal. V
53	0	53	226.82	85.19	210.21	-478.00	Rt. Knee Crease
54	0	54	203.76	4.72	203.71	-470.00	Rt. Femoral Lateral Epicn
55	0	55	133.32	74.26	110.72	-464.00	Rt. Femoral Medial Epicn
56	0	56	212.01	-55.59	204.59	-974.00	Rt. Metatarsal-Phal. V
57	0	57	226.23	67.03	216.07	-926.00	Rt. Lateral Malleolus
58	0	58	166.85	93.14	138.43	-902.00	Rt. Medial Malleolus
59	0	59	164.58	88.58	138.71	-922.00	Rt. Sphyrion
60	0	60	107.59	-41.08	99.44	-970.00	Rt. Metatarsal-Phal. I
61	0	61	251.45	138.16	210.09	-964.00	Rt. Calcaneous, Post.
62	0	62	182.96	-139.52	118.35	-984.00	Rt. Digit II
63	0	63	211.77	203.02	60.26	-474.00	Lt. Knee Crease
64	0	64	185.46	184.28	-20.85	-468.00	Lt. Femoral Lateral Epicn
65	0	65	124.47	107.21	63.23	-472.00	Lt. Femoral Medial Epicn
66	0	66	190.69	165.79	-94.21	-972.00	Lt. Metatarsal-Phal. V
67	0	67	213.13	212.76	12.62	-916.00	Lt. Lateral Malleolus
68	0	68	162.65	150.86	60.81	-904.00	Lt. Medial Malleolus
69	0	69	167.17	156.18	59.62	-918.00	Lt. Sphyrion
70	0	70	88.71	67.29	-57.81	-968.00	Lt. Metatarsal-Phal. I
71	0	71	250.17	236.03	82.92	-962.00	Lt. Calcaneous, Post.
72	0	72	167.06	58.42	-156.51	-986.00	Lt. Digit II
73	0	0	0.0	85.62	95.135	-158	Crotch

END =

APPENDIX F: COMMANDS: FUNCTIONALITY FOR HEAD AND WHOLE BODY IMAGE DATA

FUNCTIONALITY FOR HEAD AND WHOLE BODY IMAGE DATA

COMMAND	Works With Cylindrical Grid?	Works With Triangular Mesh?
<object number>	Yes	Yes
!	Yes	Yes
$	Yes	Yes
@	Yes	Yes
abssub	Yes	**No**
add	Yes	**No**
add_to_land	Yes	Yes
addobj	Yes	**No**
align	Yes	Yes
alt_land	Yes	Yes
and	Yes	**No**
auto_jaws	**No**	Yes
avgland	Yes	Yes
axes	Yes	Yes
back	Yes	Yes
balltest	Yes	**No**
black	Yes	Yes
bottom	Yes	Yes
bottom_cap	Yes	**No**
boxes	Yes	Yes
calipers	**No**	Yes
cd	Yes	Yes
center	Yes	Yes
centroid	Yes	Yes
circumference	Yes	Yes
clearance	**No**	Yes
cload/cloads	Yes	**No**
colors	Yes	Yes
comment	Yes	Yes
conclose	Yes	Yes
contour	Yes	Yes
contours	Yes	Yes
copy	Yes	**No** (USE movie_seg -)
copyland	Yes	Yes
copyseg	Yes	**No**
cursor	Yes	**No**
cwrite	Yes	**No**
cybermovie	Yes	**No**
delete	Yes	Yes
delland	Yes	Yes
delpnt	Yes	**No**
delseg	Yes	**No**
delta	Yes	**No**

derive	Yes	Yes
diff	Yes	**No**
dilate	Yes	**No**
dir	Yes	Yes
disjoint	**No**	Yes
displace	**No**	Yes
distance	Yes	Yes
do (fill/smooth)	Yes	**No**
drawline	Yes	Yes
erode	Yes	**No**
exit	Yes	Yes
eye	Yes	Yes
eyedist	Yes	Yes
fcmod	Yes	Yes
fcwrite	Yes	Yes
fill	Yes	**No**
filter	Yes	**No**
filtseg	Yes	**No**
fix_seam	Yes	**No**
fk	Yes	Yes
fkeys	Yes	Yes
force_lnd	Yes	Yes
front	Yes	Yes
fullcolor	Yes	Yes
gcv	Yes	**No**
gload	**No**	Yes
gouraud	Yes	Yes
gwrite	**No**	Yes
help	Yes	Yes
hide	Yes	Yes
histogram	Yes	**No**
interpolate	Yes	**No**
intersect	**No**	Yes
intrplnd	Yes	Yes
jaw	**No**	Yes
jump	Yes	Yes
land	Yes	Yes
landlist	Yes	Yes
left	Yes	Yes
lload	Yes	Yes
lmlist	Yes	Yes
lregister	Yes	Yes
lwrite	Yes	Yes
man	Yes	Yes
mark	Yes	**No**
median	Yes	**No**
merge	Yes	**No**
mergeSubs	**No**	Yes

mirror	Yes	Yes
mload	Yes	Yes
modland	Yes	Yes
move	Yes	Yes
move_vertex	**No**	Yes
movie_segment	**No**	Yes
mwrite	Yes	Yes
nameland	Yes	Yes
negsub	Yes	**No**
new_meas	Yes	Yes
new_order	Yes	Yes
new_vertex	**No**	Yes
newcenter	Yes	Yes
newland	Yes	Yes
open	Yes	Yes
option	Yes	Yes
ortho	Yes	Yes
pause	Yes	Yes
pick	Yes	Yes
pickmode	Yes	Yes
planes	Yes	Yes
pload	**No**	Yes
points	Yes	Yes
pop	Yes	Yes
possub	Yes	**No**
print	Yes	Yes
pshrink	**No**	Yes
push	Yes	Yes
pwrite	**No**	Yes
readout	**No**	Yes
recolor	Yes	**No**
refresh	Yes	Yes
remark	Yes	Yes
resample	Yes	Yes
rgb	Yes	Yes
right	Yes	Yes
rotate	Yes	Yes
ruin	Yes	**No**
save	Yes	Yes
scwrite	Yes	**No**
select	Yes	Yes
set	Yes	**No**
shade	Yes	**No**
show	Yes	Yes
show_hid_lnd	Yes	Yes
shrink	**No**	Yes
side	Yes	Yes
skip	Yes	Yes

skipto	Yes	Yes
sleep	Yes	Yes
smooth	Yes	**No**
split	Yes	**No**
store	Yes	Yes
subject	Yes	Yes
super	Yes	Yes
surface	Yes	Yes
surface_area	Yes	Yes
surf_register	**No**	Yes
tape	Yes	Yes
text	Yes	Yes
thin	Yes	**No**
threshold	Yes	**No**
top	Yes	Yes
toupee	Yes	**No**
transparent	Yes	Yes
trim	Yes	**No**
volume	Yes	Yes
wait	Yes	Yes
walls	Yes	Yes
white	Yes	Yes
wireframe	Yes	Yes
wload	**No**	Yes
wwrite	**No**	Yes
xload	Yes	**No**
xwrite	Yes	**No**
zload	Yes	Yes
zregister	Yes	Yes
zwrite	Yes	Yes

APPENDIX G: FILE FORMATS: DESCRIPTION OF IMAGE DATA FILES SUPPORTED BY INTEGRATE VERSION 2.8

File Formats Supported By INTEGRATE Version 2.8

I. Cyberware Scanner Format (new) (all except WB-series scanners)

ASCII header with lines of the form <keyword>=<value>\n
Header terminates with DATA=\n

Binary cylindrical data, 2 bytes per radius. First radius is lon0 lat0. 2nd radius is lon0 lat1. Typically 512 longitudes and 256 latitudes, but header (NLG, NLT) is final authority. Each radius is multiplied by 2 ** RSHIFT (value from header). RSHIFT is typically either 3 or 5. Resulting radius value is in microns, so for an RSHIFT of 3, the radius is multiplied by .008 to get millimeters. Longitude proceeds clockwise (viewed from top) and latitude goes from bottom to top.

II. Movie.byu .g format

ASCII file with 4 sections:

A. Counts Line
<npart nvert npoly nedge>

npart - number of different parts in scene,
nvert - number of vertex points in scene,
npoly - number of unique polygons in scene,
nedge - total number of polygon edges in scene

B. Part-scene Definitions
<poly1first poly1last>
poly1first - index of first polygon in first scene part (lowest = 1)
poly1last - index of last polygon in first scene part (highest = npoly)
. . .
<polyNfirst polyNlast> (N = npoly)
There are a total of npart lines in this section, or 2 * npart indexes.

C. Vertex Point Coordinates (x, y, z), 3 coordinates per Vertex, 1 or more Vertices per line.

There are a total of nvert Vertexs (or nvert * 3 coordinates) listed in this section.

D. Polygon Definitions

Polygons are defined by their vertexes. Edges are implied between adjacent vertexes in the list, and between the last vertex of a polygon and the first vertex in the polygon. The index of the last vertex of a polygon is indicated by negating the index. The lowest vertex index is 1. The largest is nvert. There are a total of nedge vertexes listed in this section. There should be a total of npoly negative vertexes (implying separate polygon definitions).

Example: 1 2 -3 defines a triangle using vertices 1, 2, and 3

III. Wavefront .obj format

Refer to Wavefront documentation/description.

IV. Two variants of Stanford .ply format

This complex format is documented at the Stanford Web Site. The formats used are a grid format ("raw .ply"; rarely used) and a vertex/face format (normally used).

V. CARD Lab ASCII point format (grid and non-grid)

1. Grid format:

```
GRID <total_points> <nlon> <nlat> <lon1>:<lonn> <lat1>:<latn>
<lon lat radius x y z>
. . .
<lon lat radius x y z>
```

2. Non-Grid format:

```
CART <total_points>
<x y z>
. . .
<x y z>
```

3. CARD Lab contour file format:

```
CONT <total_points> 0 0 0:0 0:0
<x y z> [coordinates for 1st point in contour]
. . .
<x y z> [coordinates for last point in contour]
```

VI. Old CARD Lab Landmark File Format

```
5 1 1
<lon lat radius x y z> for standard landmark 1
```

```
. . .
5 <N> 1
<lon lat radius x y z> for standard landmark 'N'
-1 -1 -1 [end of file tag]
```

VII. New CARD Lab Landmark File Format

```
SUBJECT_ID = <subject identifier>
SCAN_TYPE  = <type identifier>
STUDY_NAME = <acquisition study identifier>
LAND_STUDY = <landmark study identifier>
STD_LAND = 42
AUX_LAND = <N> [number of auxiliary landmarks in dataset]
STANDARD =
 1 <lon> <lat> <radius> <x> <y> <z>
. . .
42 <lon> <lat> <radius> <x> <y> <z>
AUX =
 1 <lon> <lat> <radius> <x> <y> <z> <auxiliary landmark 1 name>
. . .
[N] <lon> <lat> <radius> <x> <y> <z> <auxiliary landmark N name>
END =
```

VIII. CARD Lab matrix file format

```
<original file name>
<original subject id or duplicate of file name>
<x1> <y1> <z1> <t1> [displacement matrix]
<x2> <y2> <z2> <t2>
<x3> <y3> <z3> <t3>
<x4> <y4> <z4> <t4>
THIN <lon thin> <lat thin> [sub-sampling intervals on lon and
lat]
TRIM <low lon> <high lon> <low lat> <high lat> [include bounds on
lon and lat]
CENTER <x> <y> <z> [displacement(s) from original object center]
```

APPENDIX H: DEFINITIONS FOR DETERMINATION OF JOINT CENTERS

Definitions For Determination of Joint Centers

Ankles, right and left: use midpoint between Lateral Malleolus and Sphyrion

Knees, right and left: use midpoint between Lateral and Medial Femoral Epicondyles

Hips, right and left:
1) start at midpoint between Anterior Superior Iliac Spine and Symphysion
2) translate in the posterior direction to the plane of the Trochanterions
3) translate 15 mm down

Pelvic Joint:
1) start at Posterior Superior Iliac Midspine coordinates
2) translate 51 mm in the anterior direction

Abdomen Joint:
1) start at 10^{th} Rib Midspine coordinates
2) translate 51 mm in the anterior direction

Thorax Joint:
1) start at Cervicale coordinates
2) translate 51 mm in the anterior direction
3) translate 25 mm down

Head/Neck Joint: use midpoint between right and left Tragions

Shoulder, right and left:
1) start at Acromion coordinates
2) translate 38 mm in the medial direction
3) translate 38 mm down

Elbow, right and left: use midpoint between Medial and Lateral Humeral Epicondyles

Wrist, right or left: use midpoint between Radial and Ulnar Styloid Processes

APPENDIX I: SYNTAX FOR SCRIPT FILE MATHEMATICAL EXPRESSIONS

Syntax for Mathematical Expressions Used in Scripts

INTEGRATE supports a limited ability to apply math within scripts. An opening parenthesis ('(') begins a math expression and a closing parenthesis (')') ends the expression. Expressions may be nested as necessary. Within an expression, the standard math operators +, -, *, /, are supported. It is possible to access some information about an object as part of a math expression. Landmark coordinates, extreme values in the 3 axes, and the offset of the object from its native axis system can be used in scripts.

A math expression has either the syntax (operand operator operand) or the syntax (operand) in all cases. The operator is one of the 4 above or is either MIN (minimum of the 2 operands) or MAX (maximum of the 2 operands).

Operands can be either a numeric constant or an object parameter. Object parameters have the following syntax:

&<obj#><obj param type><landmark# as applicable><coordinate axis for value>

Available object parameter types are:
- standard landmark coordinates (L),
- auxiliary landmark coordinates (Z),
- object position coordinates (P),
- object maximum value along an axis (T), or
- object minumum value along an axis (B).

EXAMPLES:

```
# move object so that obj 1 aux land 25 is on XY plane
move 0 0 (0 - &1Z25Z)
# remove points below obj 1 aux land 25 and outside aux land 77 and 78
movie_seg ly(&1Z25Y) lx(&1Z77X) ux(&1Z78X)
# create a new landmark at obj 1 aux land 10 unless aux land 12 lower
newland z88 (&1Z10X) (&1Z10Y$MIN$&1Z12Y) (&1Z10Z)
# create landmarks at top and bottom center of object 1
newland z89 0 (&1TY) 0
newland z90 0 (&1BY) 0
```